

Progetto Basi di Dati e di Conoscenza:

**“Stagionalità Cibi”**



Zagaria Simone

0282174

A.A. 2021/2022

Docente: Loredana Vigliano

INDICE

Sommario

[Introduzione 4](#_Toc126529916)

[Introduzione al database 4](#_Toc126529917)

[Glossario 4](#_Toc126529918)

[Analisi dei Requisiti 5](#_Toc126529919)

[Descrizione del Problema 5](#_Toc126529920)

[Analisi delle Entità 6](#_Toc126529921)

[Progettazione Concettuale, Logica e Fisica 15](#_Toc126529922)

[Progettazione Concettuale 15](#_Toc126529923)

[Progettazione Logica 16](#_Toc126529924)

[Progettazione Fisica 17](#_Toc126529925)

[Query 17](#_Toc126529926)

[Query SQL 18](#_Toc126529927)

[Query 1 18](#_Toc126529928)

[Query 2 18](#_Toc126529929)

[Query 3 18](#_Toc126529930)

[Query 4 19](#_Toc126529931)

[Query 5 19](#_Toc126529932)

[Query 6 19](#_Toc126529933)

[Query 7 20](#_Toc126529934)

[Query 8 20](#_Toc126529935)

[Query 9 20](#_Toc126529936)

[Query 10 21](#_Toc126529937)

[Query 11 21](#_Toc126529938)

[Query 12 21](#_Toc126529939)

[Query 13 21](#_Toc126529940)

[Query 14 22](#_Toc126529941)

[Query 15 22](#_Toc126529942)

[Query 16 22](#_Toc126529943)

[Query 17 22](#_Toc126529944)

[Query 18 23](#_Toc126529945)

[Query 19 23](#_Toc126529946)

[Query in Calcolo Relazionale 23](#_Toc126529947)

[Query 3 23](#_Toc126529948)

[Query 18 23](#_Toc126529949)

[Ottimizzazione Query 24](#_Toc126529950)

[Trigger, Stored Procedures e Views 24](#_Toc126529951)

[Trigger 24](#_Toc126529952)

[Before\_insert\_anno\_acquistoterreno 24](#_Toc126529953)

[Before\_update\_anno\_acquistoterreno 25](#_Toc126529954)

[Before\_insert\_durata\_diete 25](#_Toc126529955)

[Before\_update\_durata\_diete 26](#_Toc126529956)

[Stored Procedures 26](#_Toc126529957)

[Nuovo\_utente 26](#_Toc126529958)

[Nuova\_dieta 27](#_Toc126529959)

[Inserisci\_fine\_dieta 27](#_Toc126529960)

[Views 28](#_Toc126529961)

[ProdottiStagionali\_PerMese 28](#_Toc126529962)

[Alimenti\_PerDieta 28](#_Toc126529963)

[Funzioni\_Vitamine 28](#_Toc126529964)

[Funzioni\_SaliMinerali 29](#_Toc126529965)

[Sicurezza 29](#_Toc126529966)

[INSERT 30](#_Toc126529967)

[Mese 30](#_Toc126529968)

[Vitamina 30](#_Toc126529969)

[SaleMinerale 30](#_Toc126529970)

[Stagionalità 32](#_Toc126529971)

[ContieneSM 36](#_Toc126529972)

[ContieneV 39](#_Toc126529973)

[Consorzio 43](#_Toc126529974)

[Terreno 43](#_Toc126529975)

[Coltivazione 43](#_Toc126529976)

[Funzione 45](#_Toc126529977)

[SMSvolge 45](#_Toc126529978)

[VSvolge 46](#_Toc126529979)

[Dieta 46](#_Toc126529980)

[VUsataIn 46](#_Toc126529981)

[SMUsatoIn 47](#_Toc126529982)

[EffettuataDa 47](#_Toc126529983)

# Introduzione

## Introduzione al database

Stagionalità Cibi è un database che nasce con lo scopo di enfatizzare l’importanza del consumo di prodotti di stagione, che può portar a vari benefici, tra i quali:

* Valori nutrizionali più alti
* Gusto migliore
* Risparmio economico
* Minore impatto ambientale

Il database mantiene per ogni frutta e verdura il periodo dell’anno in cui sono considerati di stagione, i valori nutrizionali, i sali minerali e le vitamine corrispondenti ad ognuno di questi alimenti. Per ogni sale minerale e vitamina sono anche indicate le funzioni che svolgono all’interno del nostro corpo.

Inoltre, tiene conto di varie diete effettuabili a partire da questi nutrienti e delle persone che hanno deciso di sottoporsi a queste diete, in modo da poter consigliare loro la frutta e la verdura più adatta per poter soddisfare il fabbisogno giornaliero di nutrienti richiesto dalla loro dieta.

Infine, vengono presi in considerazione anche i terreni in cui vengono prodotti questi alimenti e i consorzi agrari che possiedono questi terreni.

## Glossario

|  |  |
| --- | --- |
| Termine | Definizione |
| Consorzio agrario | Unione di imprese di natura agricola volta a massimizzare i profitti delle aziende stesse tramite la mutua collaborazione |
| Dieta | Definisce la quantità giornaliera di sali minerali e vitamine che una persona deve introdurre nella propria alimentazione per un certo arco di tempo per ricevere determinati benefici |
| Macroelementi | Sottotipo di sali minerali presenti in quantità discrete nel nostro organismo. Il fabbisogno giornaliero supera i 100mg |
| Microelementi | Sottotipo di sali minerali presenti solo in tracce nel nostro organismo. Il fabbisogno giornaliero è limitato e non supera i 100mg |
| Nutriente | Sostanza presente nei cibi, indispensabile per la vita e per la normale funzionalità fisiologica e metabolica di tutti gli organismi viventi |
| Sale minerale | Nutrienti di natura inorganica indispensabili per il corretto funzionamento del nostro organismo |
| Stagionalità | Ritmo naturale che caratterizza i periodi in cui frutta e verdura nascono e si sviluppano fino alla maturità. Tutti i frutti e le verdure crescono in maniera ottimale soltanto in periodi specifici dell’anno, a seconda delle stagioni |
| Valori nutrizionali | Informazioni sulla composizione chimica di un alimento |
| Vitamina | Nutriente di natura organica necessario come agente regolatore di varie funzioni dell’organismo. |
| Vitamina idrosolubile | Sottotipo di vitamine solubili in acqua. Non sono accumulabili nel nostro organismo e sono quindi da assumere quotidianamente con l’alimentazione |
| Vitamina liposolubile | Sottotipo di vitamine solubili in olio e grassi. Grazie alla loro affinità ai grassi, possono depositarsi nel tessuto adiposo e quindi il nostro corpo ne può mantenere una riserva |

# Analisi dei Requisiti

## Descrizione del Problema

Nel Database “Stagionalità Cibi” si vuole poter memorizzare vari tipi di frutta e verdura e, per ognuno di essi, si vogliono sapere:

* Informazioni di base
* Periodo di stagionatura
* Apporti vitaminici
* Sali minerali presenti e quantità

Si vuole inoltre poter specificare le funzioni che svolgono nel nostro organismo ciascun sale minerale e vitamina e poter creare diete a partire da questi nutrienti.

Ogni dieta deve poter contenere:

* Nome e una breve descrizione
* Un insieme di vitamine e/o Sali minerali
* Un apporto giornaliero per ognuno di questi ultimi

Le diete devono poter essere effettuabili da utenti che si possono registrare al database.   
Per ogni utente registrato si vogliono mantenere:

* Nome
* Cognome
* Data di nascita
* Telefono
* Indirizzo
* Email (facoltativa)
* Diete effettuate
* Data di inizio di queste diete
* Data di termine di queste diete (se terminate)

Ogni utente deve poter partecipare a quante diete vuole e deve poter vedere quali sono i cibi di stagione che soddisfano il fabbisogno giornaliero di quella dieta.

Infine, si vogliono poter registrare i consorzi agricoli e i terreni che possiedono, in modo che gli utenti possano sapere esattamente le aziende e i campi in cui vengono prodotti gli alimenti di stagione che a loro interessano.

Per ogni Consorzio si vuole memorizzare:

* Nome
* Partita IVA
* Sede legale
* Recapito telefonico
* Anno di fondazione
* Terreni posseduti

Mentre per ogni terreno si vuole sapere:

* Indirizzo
* Superficie in mq
* Anno in cui è stato acquistato
* Numero di dipendenti (facoltativo)
* Prodotti che coltivano

## Analisi delle Entità

ProdottoTerra

L’entità ***ProdottoTerra*** mantiene per ogni alimento i nomi comuni e scientifici e la quantità di energia in Kcal che contiene in 100 grammi. Gli alimenti inseriti possono essere solo frutta o verdura, e questo è specificato attraverso una constraint sull’attributo tipo.

È stato aggiunto l’attributo kcal in questa entità perché non si tratta di un nutriente ma del valore energetico che ogni alimento possiede e quindi un attributo che caratterizza l’alimento.

CREATE TABLE ProdottoTerra (

id\_prodottoterra VARCHAR(7) NOT NULL,

nomecomune\_prodottoterra VARCHAR(50) NOT NULL,

nomescientifico\_prodottoterra VARCHAR(50) NOT NULL,

kcal\_100g INT NOT NULL,

tipo VARCHAR(10) NOT NULL,

PRIMARY KEY (id\_prodottoterra,

CHECK (tipo = 'frutta' OR tipo = 'verdura')

);

Mese

L’entità ***Mese*** contiene i nomi di tutti i mesi italiani e la stagione relativa ad ogni mese. È stato scelto di mettere come chiave primaria il nome del mese dato che non esistono due mesi con lo stesso nome.

CREATE TABLE Mese (

nome\_mese VARCHAR(20) NOT NULL,

stagione\_mese VARCHAR(20) NOT NULL,

PRIMARY KEY (nome\_mese)

);

Stagionalità

L’entità ***Stagionalità*** è generata dalla relazione n a n tra *ProdottoTerra* e *Mese* e serve ad associare ogni prodotto ai mesi in cui è considerato “di stagione”. Infatti, ogni prodotto può stagionare in più mesi e in ogni mese possono stagionare più prodotti.

CREATE TABLE Stagionalità (

nome\_mese VARCHAR(20) NOT NULL,

id\_prodottoterra VARCHAR(7) NOT NULL,

PRIMARY KEY (nome\_mese, id\_prodottoterra),

FOREIGN KEY (id\_prodottoterra) REFERENCES ProdottoTerra(id\_prodottoterra) ON DELETE NO ACTION ON UPDATE CASCADE,

FOREIGN KEY (nome\_mese) REFERENCES Mese(nome\_mese) ON DELETE NO ACTION ON UPDATE CASCADE

);

Vitamina

L’entità ***Vitamina*** contiene le informazioni generali per ogni vitamina conosciuta e la tipologia a cui appartiene attraverso l’attributo tipo (infatti le vitamine possono essere o idrosolubili o liposolubili). È stato scelto di mettere il nome comune delle vitamine come chiave primaria dato che non possono esistere due vitamine chiamate con lo stesso nome.

CREATE TABLE Vitamina (

nome\_comunevitamina VARCHAR(10) NOT NULL,

nome\_chimicovitamina VARCHAR(50) NOT NULL,

tipo VARCHAR(30) NOT NULL,

PRIMARY KEY (nome\_comunevitamina),

CHECK (tipo = 'idrosolubile' OR tipo = 'liposolubile')

);

ContieneV

L’entità ***ContieneV*** è generata dalla relazione n a n tra *ProdottoTerra* e *Vitamina* ed indica, per ogni prodotto, le vitamine che contiene e in che quantità. Infatti, ogni prodotto può contenere da 0 a n vitamine e ogni vitamina può esser contenuta da 0 a n prodotti tra quelli nella tabella ProdottoTerra.

CREATE TABLE ContieneV (

id\_prodottoterra VARCHAR(7) NOT NULL,

nome\_comunevitamina VARCHAR(10) NOT NULL,

quantitàper100g FLOAT NOT NULL,

PRIMARY KEY (id\_prodottoterra, nome\_comunevitamina),

FOREIGN KEY (id\_prodottoterra) REFERENCES ProdottoTerra (id\_prodottoterra) ON DELETE NO ACTION ON UPDATE CASCADE,

FOREIGN KEY (nome\_comunevitamina) REFERENCES Vitamina (nome\_comunevitamina) ON DELETE NO ACTION ON UPDATE CASCADE

);

SaleMinerale

L’entità ***SaleMinerale*** contiene le informazioni generali per ogni sale minerale rilevante all’alimentazione e la tipologia a cui appartiene attraverso l’attributo tipo (infatti i Sali minerali possono essere o macroelementi o microelementi). È stato scelto di mettere il nome chimico del sale minerale come chiave primaria dato che nella tavola periodica non possono esistere due elementi con simboli chimici uguali.

CREATE TABLE SaleMinerale (

simbolo\_elementochimico VARCHAR(2) NOT NULL,

nome\_elementochimico VARCHAR(50) NOT NULL,

tipo VARCHAR(30) NOT NULL,

PRIMARY KEY (simbolo\_elementochimico),

CHECK (tipo = 'macroelemento' OR tipo = 'microelemento')

);

ContieneSM

L’entità ***ContieneSM***è generata dalla relazione n a n tra *ProdottoTerra* e *SaleMinerale* e serve ad indicare, per ogni prodotto, i Sali minerali che contiene e in che quantità. Infatti, ogni prodotto può contenere da 0 a n Sali minerali e ogni Sale minerale può esser contenuto da 0 a n prodotti tra quelli nella tabella ProdottoTerra.

CREATE TABLE ContieneSM (

id\_prodottoterra VARCHAR(7) NOT NULL,

simbolo\_elementochimico VARCHAR(2) NOT NULL,

quantitàper100g FLOAT NOT NULL,

PRIMARY KEY (id\_prodottoterra, simbolo\_elementochimico),

FOREIGN KEY (id\_prodottoterra) REFERENCES ProdottoTerra(id\_prodottoterra) ON DELETE NO ACTION ON UPDATE CASCADE,

FOREIGN KEY (simbolo\_elementochimico) REFERENCES SaleMinerale(simbolo\_elementochimico) ON DELETE NO ACTION ON UPDATE CASCADE

);

Consorzio

L’entità ***Consorzio*** contiene le informazioni richieste dalla descrizione del problema per i vari consorzi agrari e usa la propria partita IVA come chiave primaria.

CREATE TABLE Consorzio (

partitaIVA INT NOT NULL,

nome\_consorzio VARCHAR(100) NOT NULL,

sede\_legale VARCHAR(50) NOT NULL,

anno\_fondazione YEAR NOT NULL,

telefono VARCHAR(13),

PRIMARY KEY (partitaIVA)

);

Terreno

L’entità ***Terreno*** contiene per ogni terreno le informazioni richieste dalla descrizione del problema. Inoltre, contiene l’attributo partita IVA, una Foreign key che fa riferimento al consorzio proprietario del terreno e le informazioni riguardo il possesso del terreno (l’anno di acquisto e il numero di dipendenti). Infatti, ogni terreno può esser posseduto da un solo consorzio, mentre ogni consorzio può possedere più terreni (relazione 1 a n);

CREATE TABLE Terreno (

id\_terreno VARCHAR(10) NOT NULL,

indirizzo VARCHAR(100) NOT NULL,

mq\_terreno INT NOT NULL,

anno\_acquistoterreno YEAR NOT NULL,

numero\_dipendenti INT,

partitaIVA INT NOT NULL,

PRIMARY KEY (id\_terreno),

FOREIGN KEY (partitaIVA) REFERENCES Consorzio(partitaIVA) ON DELETE NO ACTION ON UPDATE CASCADE

);

Coltivazione

L’entità ***Coltivazione***è generata dalla relazione n a n tra *Terreno* e *ProdottoTerra* e indica in ogni terreno che prodotti vengono coltivati. Infatti, ogni prodotto può esser coltivato in più terreni e ogni terreno può coltivare da 1 a n prodotti tra quelli nella tabella ProdottoTerra.

CREATE TABLE Coltivazione (

id\_terreno VARCHAR(10) NOT NULL,

id\_prodottoterra VARCHAR(7) NOT NULL,

PRIMARY KEY (id\_terreno, id\_prodottoterra),

FOREIGN KEY (id\_prodottoterra) REFERENCES ProdottoTerra(id\_prodottoterra) ON DELETE NO ACTION ON UPDATE CASCADE,

FOREIGN KEY (id\_terreno) REFERENCES Terreno (id\_terreno) ON DELETE NO ACTION ON UPDATE CASCADE

);

Funzione

L’entità ***Funzione*** contiene tutte le possibili funzioni che possono avere i nutrienti (cioè in questo caso sali minerali e vitamine) e una breve descrizione (se necessaria) per ognuna di esse

CREATE TABLE Funzione (

id\_funzione VARCHAR(7) NOT NULL,

nome\_funzione VARCHAR(50) NOT NULL,

descrizione MEDIUMTEXT,

PRIMARY KEY (id\_funzione)

);

VSvolge

L’entità ***VSvolge***è generata dalla relazione n a n tra *Vitamina* e *Funzione* e serve ad indicare, per ogni Vitamina, le funzioni che svolge nel nostro organismo. Infatti, ogni Vitamina può avere da 1 a n funzioni e ogni funzione può esser svolta da 1 a n Vitamine.

CREATE TABLE VSvolge (

nome\_comunevitamina VARCHAR(10) NOT NULL,

id\_funzione VARCHAR(7) NOT NULL,

PRIMARY KEY (nome\_comunevitamina, id\_funzione),

FOREIGN KEY (nome\_comunevitamina) REFERENCES Vitamina(nome\_comunevitamina) ON DELETE NO ACTION ON UPDATE CASCADE,

FOREIGN KEY (id\_funzione) REFERENCES Funzione(id\_funzione) ON DELETE NO ACTION ON UPDATE CASCADE

);

SMSvolge

L’entità ***SMSvolge***è generata dalla relazione n a n tra *SaleMinerale* e *Funzione* e serve ad indicare, per ogni Sale minerale, le funzioni che svolge nel nostro organismo. Infatti, ogni Sale minerale può avere da 1 a n funzioni e ogni funzione può esser svolta da 1 a n Sali minerali.

CREATE TABLE SMSvolge (

simbolo\_elementochimico VARCHAR(2) NOT NULL,

id\_funzione VARCHAR(7) NOT NULL,

PRIMARY KEY (simbolo\_elementochimico, id\_funzione),

FOREIGN KEY (simbolo\_elementochimico) REFERENCES SaleMinerale(simbolo\_elementochimico) ON DELETE NO ACTION ON UPDATE CASCADE,

FOREIGN KEY (id\_funzione) REFERENCES Funzione(id\_funzione) ON DELETE NO ACTION ON UPDATE CASCADE

);

Dieta

L’entità ***Dieta*** contiene, per ogni dieta, il nome, una durata raccomandata ed una breve descrizione facoltativa per ognuna di esse.

CREATE TABLE Dieta (

id\_dieta VARCHAR(10) NOT NULL,

nome\_dieta VARCHAR(50) NOT NULL,

durataraccomandata\_ingiorni VARCHAR(20) NOT NULL,

descrizione MEDIUMTEXT NULL,

PRIMARY KEY (id\_dieta)

);

VUsataIn

L’entità ***VUsataIn***è generata dalla relazione n a n tra *Dieta* e *Vitamina* e serve ad indicare, per ogni Dieta, quali vitamine raccomanda e in che quantità giornaliera. Le quantità di vitamine raccomandate nelle diete sono per adulti dai 18 anni in su. Ogni dieta può contenere da 0 a n vitamine e ogni vitamina può esser contenuta in da 0 a n diete.

CREATE TABLE VUsataIn (

id\_dieta VARCHAR(10) NOT NULL,

nome\_comunevitamina VARCHAR(10) NOT NULL,

quantitàgiornaliera\_ingrammi FLOAT NOT NULL,

PRIMARY KEY (id\_dieta, nome\_comunevitamina),

FOREIGN KEY (id\_dieta) REFERENCES Dieta(id\_dieta) ON DELETE NO ACTION ON UPDATE CASCADE,

FOREIGN KEY (nome\_comunevitamina) REFERENCES Vitamina(nome\_comunevitamina) ON DELETE NO ACTION ON UPDATE CASCADE

);

SMUsatoIn

L’entità ***SMUsatoIn***è generata dalla relazione n a n tra *Dieta* e *SaleMinerale* e serve ad indicare, per ogni Dieta, quali Sali minerali raccomanda e in che quantità giornaliera. Le quantità di Sali minerali raccomandate nelle diete sono per adulti dai 18 anni in su. Ogni dieta può contenere da 0 a n Sali minerali e ogni sale minerale può esser contenuto in da 0 a n diete.

CREATE TABLE SMUsatoIn (

id\_dieta VARCHAR(10) NOT NULL,

simbolo\_elementochimico VARCHAR(7) NOT NULL,

quantitàgiornaliera\_ingrammi FLOAT NOT NULL,

PRIMARY KEY (id\_dieta, simbolo\_elementochimico),

FOREIGN KEY (id\_dieta) REFERENCES Dieta(id\_dieta) ON DELETE NO ACTION ON UPDATE CASCADE,

FOREIGN KEY (simbolo\_elementochimico) REFERENCES SaleMinerale (simbolo\_elementochimico) ON DELETE NO ACTION ON UPDATE CASCADE

);

Persona

L’entità ***Persona*** contiene le informazioni generali richieste dal problema per ogni persona registrata al database.

CREATE TABLE Persona (

CF CHAR(16) NOT NULL,

nome VARCHAR(50) NOT NULL,

cognome VARCHAR(50) NOT NULL,

data\_nascita DATE NOT NULL,

indirizzo VARCHAR(100) NOT NULL,

telefono VARCHAR(13) NOT NULL,

email VARCHAR(50) NULL,

PRIMARY KEY (CF)

);

EffettuataDa

L’entità ***EffettuataDa***è generata dalla relazione n a n tra *Persona* e *Dieta* mantiene per ogni persona le diete effettuate. Per ognuna abbiamo anche una data d’inizio e un’eventuale data di fine. Infatti, ogni persona può effettuare da 0 a n diete e ogni dieta può esser effettuata da 0 a n persone. Come chiave primaria si è scelta la tripla (CF,id\_dieta,inizio\_dieta) dato che ogni dieta può esser ripetuta più volte nel tempo, ma non è possibile iniziare due volte la stessa dieta nello stesso giorno.  
Infine, c’è un constraint sulle date d’inizio e di fine dieta in modo che la fine della dieta non possa avvenir prima dell’inizio.

CREATE TABLE EffettuataDa (

CF CHAR(16) NOT NULL,

id\_dieta VARCHAR(10) NOT NULL,

inizio\_dieta DATE NOT NULL,

fine\_dieta DATE,

PRIMARY KEY (CF,id\_dieta,inizio\_dieta),

FOREIGN KEY (CF) REFERENCES Persona(CF) ON DELETE NO ACTION ON UPDATE CASCADE,

FOREIGN KEY (id\_dieta) REFERENCES Dieta(id\_dieta) ON DELETE NO ACTION ON UPDATE CASCADE

);

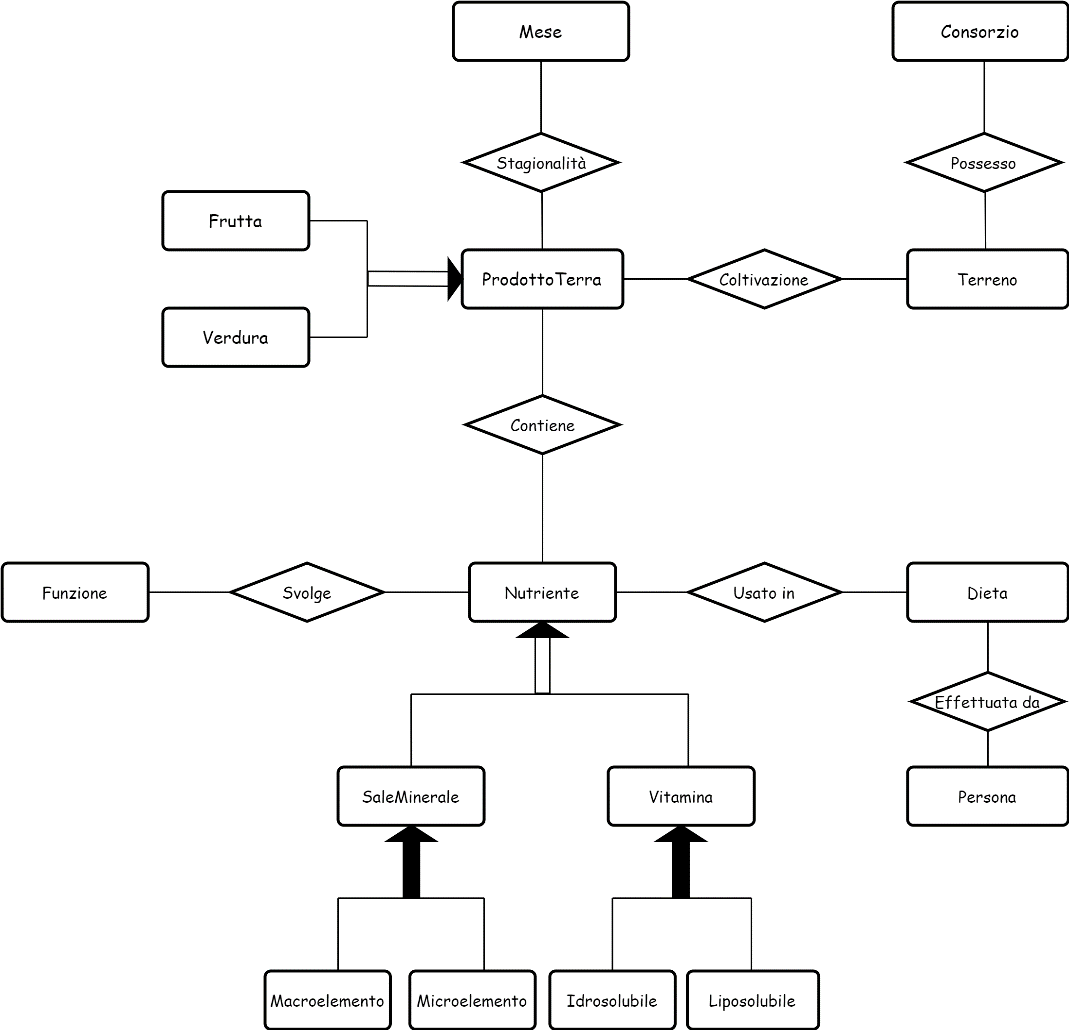
Operazioni più frequenti

Le operazioni che si stimano saranno più frequenti nel database sono:

1. Aggiunta di un nuovo utente al database
2. Inizio di una dieta
3. Termine di una dieta
4. Ricerca dei prodotti di stagione per soddisfare una certa dieta
5. Ricerca delle funzioni dei nutrienti
6. Ricerche sui valori nutritivi contenuti nei prodotti
7. Ricerca sui consorzi che vendono i prodotti

# Progettazione Concettuale, Logica e Fisica

## Progettazione Concettuale

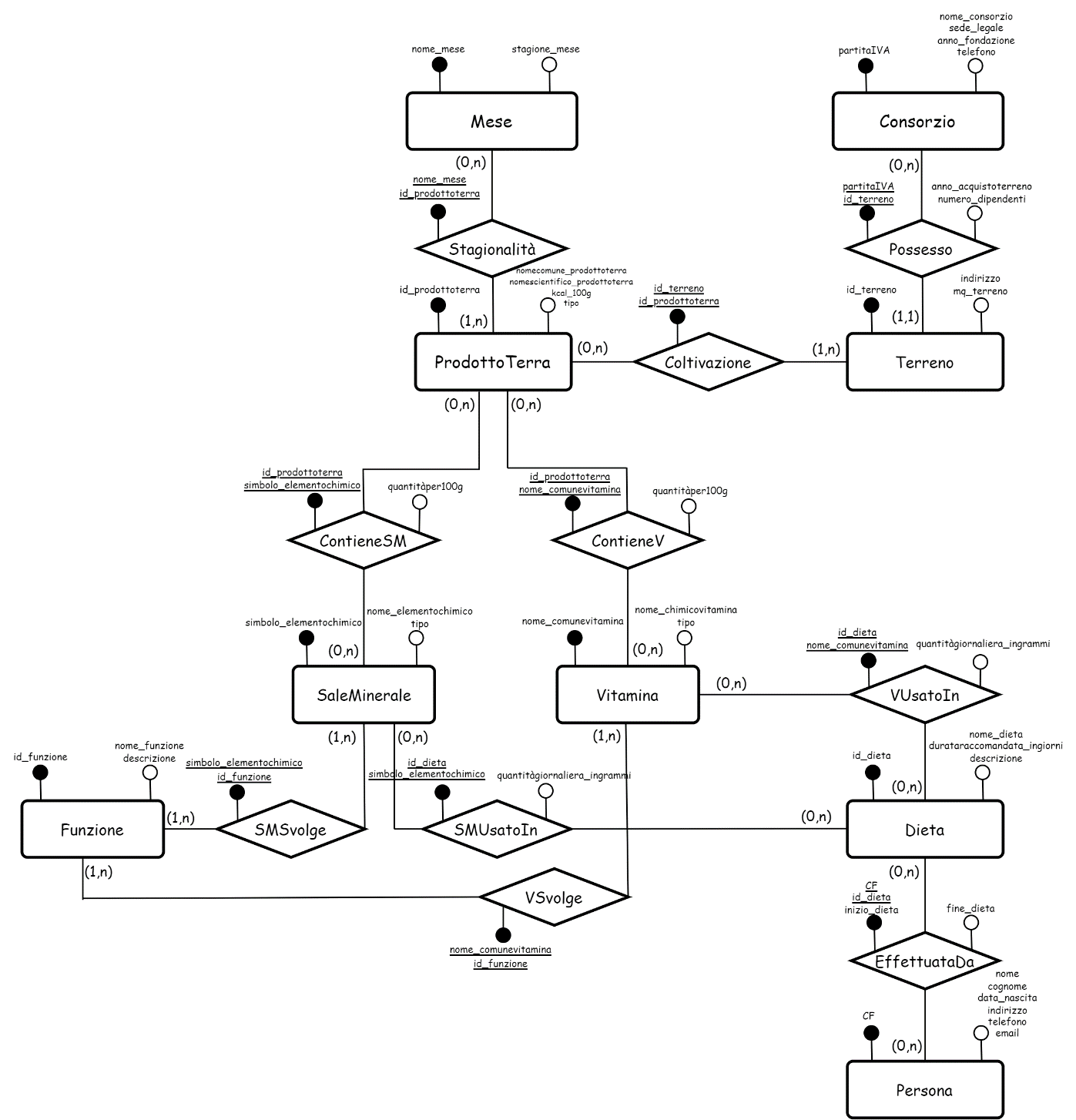


Nello schema concettuale sono presenti **quattro generalizzazioni**. Due di queste **sono parziali esclusive** (ProdottoTerra e Nutriente) e due di queste sono **totali esclusive** (SaleMinerale e Vitamina).

Si è deciso di considerare soltanto Sali Minerali e Vitamine come categorie di nutrienti in quanto sono i due maggiormente presenti in frutta e verdura.

Tutte le quantità (sia il contenuto in vitamine/Sali minerali dei prodotti, sia l’apporto giornaliero delle diete) saranno indicate in grammi.

## Progettazione Logica



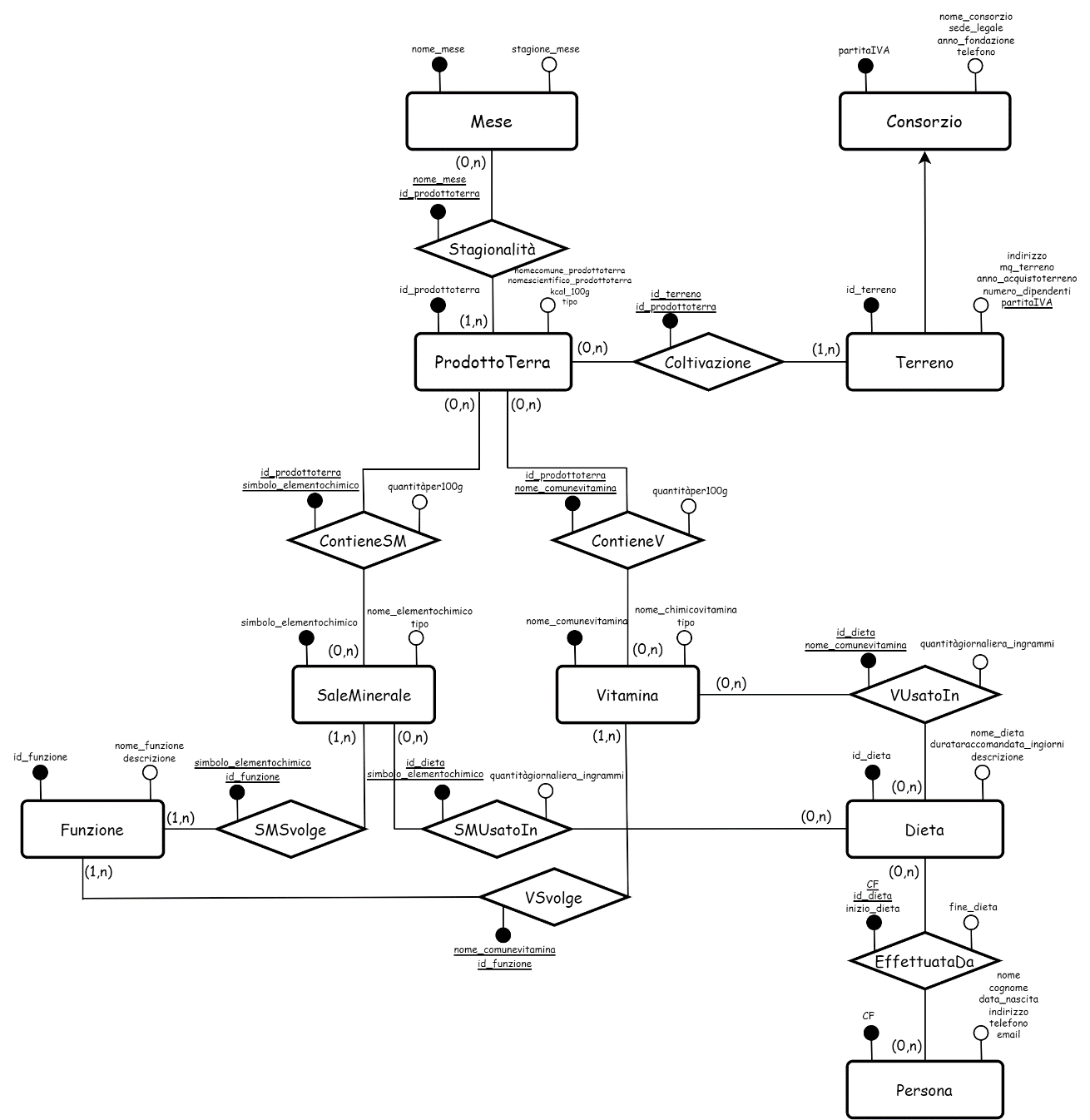
Nella progettazione logica sono stati aggiunti gli attributi e sono state risolte le generalizzazioni.

Si è deciso di accorpare le generalizzazioni ProdottoTerra, SaleMinerale e Vitamina nel padre aggiungendo l’attributo “tipo”, dato che i figli avevano pochi o nessun attributo che li caratterizzavano. Gli attributi tipo potranno assumere solo i valori dei relativi figli attraverso vincoli appositi.

Si è invece deciso per l’entità Nutriente di accorparla nei figli, dato che gli attributi di ognuno dei tipi di nutrienti sarebbero troppo diversi e accorpando le entità figlie nel padre ci sarebbero stati molti valori NULL.

Infine, è stato verificato che non esistono attributi composti e che quindi lo schema è in prima forma normale. Inoltre, risulta normalizzato per quanto riguarda la seconda e terza forma normale. Infatti, ogni attributo non primo dipende funzionalmente da ogni chiave in modo non transitivo.

## Progettazione Fisica



Tutte le relazioni eccetto Possesso sono n a n, e per quelle si è mantenuta una tabella apposita. La relazione Possesso (1 a n) invece è stata eliminata inserendo una Foreign key e gli altri attributi della relazione nell’entità che partecipa alla relazione con cardinalità massima 1, cioè Terreno.

# Query

Riporto di seguito una quindicina di query significative per il database. Gli insert sono posti in fondo al documento in quanto molto corposi

## Query SQL

### Query 1

Si vogliono sapere tutti i nutrienti e le relative quantità presenti nelle carote.

SELECT nome\_comunevitamina as nutriente, quantitàper100g

FROM ProdottoTerra PT, ContieneV CV

WHERE PT.id\_prodottoterra = CV.id\_prodottoterra AND

nomecomune\_prodottoterra = 'carote'

UNION ALL

SELECT simbolo\_elementochimico, quantitàper100g

FROM ProdottoTerra PT,ContieneSM CSM

WHERE PT.id\_prodottoterra = CSM.id\_prodottoterra AND

nomecomune\_prodottoterra = 'carote';

9 rows in set (0.01 sec)

### Query 2

Si vuole sapere la classifica delle diete più popolari tra gli utenti del database

SELECT D.nome\_dieta, count(ED.id\_dieta) as numero\_volte

FROM Dieta D,EffettuataDa ED

WHERE D.id\_dieta = ED.id\_dieta

GROUP BY D.id\_dieta

ORDER BY count(D.id\_dieta) DESC;

8 rows in set (0.01 sec)

### Query 3

Si vuole sapere i nomi di tutta la frutta che è di stagione in estate

SELECT DISTINCT PT.nomecomune\_prodottoterra

FROM ProdottoTerra PT,Stagionalità S,Mese M

WHERE PT.id\_prodottoterra = S.id\_prodottoterra AND

S.nome\_mese = M.nome\_mese AND

M.stagione\_mese = 'estate' AND

PT.tipo = 'frutta';

11 rows in set (0.01 sec)

### Query 4

Per ogni consorzio fondato prima del 1970, si vogliono sapere tutti i prodotti che produce

SELECT DISTINCT C.nome\_consorzio, PT.nomecomune\_prodottoterra

FROM Consorzio C,Terreno T,Coltivazione COL,ProdottoTerra PT

WHERE C.partitaIVA = T.partitaIVA AND

COL.id\_terreno = T.id\_terreno AND

COL.id\_prodottoterra = PT.id\_prodottoterra AND

C.anno\_fondazione <= 1970

ORDER BY nome\_consorzio ASC;  
  
27 rows in set (0.01 sec)

### Query 5

Si vogliono sapere le verdure che possiedono i nutrienti che potrebbero soddisfare la dieta contro la spossatezza

SELECT PT.nomecomune\_prodottoterra

FROM Dieta D,VUsataIn VIN,Vitamina V,ContieneV CV,ProdottoTerra PT

WHERE D.id\_dieta = VIN.id\_dieta AND

VIN.nome\_comunevitamina = V.nome\_comunevitamina AND

V.nome\_comunevitamina = CV.nome\_comunevitamina AND

CV.id\_prodottoterra = PT.id\_prodottoterra AND

D.nome\_dieta = 'dieta contro la spossatezza' AND

PT.tipo = 'verdura'

UNION

SELECT PT2.nomecomune\_prodottoterra

FROM Dieta D2, SMUsatoIn SMIN,SaleMinerale SM,ContieneSM CSM,ProdottoTerra PT2

WHERE D2.id\_dieta = SMIN.id\_dieta AND

SMIN.simbolo\_elementochimico = SM.simbolo\_elementochimico AND

SM.simbolo\_elementochimico = CSM.simbolo\_elementochimico AND

CSM.id\_prodottoterra = PT2.id\_prodottoterra AND

D2.nome\_dieta = 'dieta contro la spossatezza' AND

PT2.tipo = 'verdura';

18 rows in set (0.03 sec)

### Query 6

Il nome e il cognome delle persone che si sono sottoposte a 3 o più diete a partire dal 1/09/2022

SELECT P.nome,P.cognome

FROM Persona P, EffettuataDa ED

WHERE P.CF = ED.CF AND

ED.inizio\_dieta >= '2022-09-01'

GROUP BY P.nome,P.cognome

HAVING count(ED.id\_dieta) >= 3;

1 row in set (0.02 sec)

### Query 7

Si vuole sapere, per ogni vitamina, che funzioni ha nel nostro organismo

SELECT V.nome\_comunevitamina, F.nome\_funzione,F.descrizione

FROM Vitamina V, VSvolge VS, Funzione F

WHERE V.nome\_comunevitamina = VS.nome\_comunevitamina AND

VS.id\_funzione = F.id\_funzione;

18 rows in set (0.01 sec)

### Query 8

Si vuole conoscere l’alimento che contiene più Magnesio e la relativa quantità

SELECT PT.nomecomune\_prodottoterra,PT.nomescientifico\_prodottoterra,PT.Kcal\_100g,PT.tipo,CSM.quantitàper100g as quantitàmagnesioper100g

FROM ProdottoTerra PT, ContieneSM CSM, SaleMinerale SM

WHERE SM.simbolo\_elementochimico = CSM.simbolo\_elementochimico AND

CSM.id\_prodottoterra = PT.id\_prodottoterra AND

nome\_elementochimico = 'magnesio'

ORDER BY quantitàmagnesioper100g DESC

LIMIT 1;

1 row in set (0.02 sec)

### Query 9

Si vuole conoscere il numero di utenti che hanno meno di 40 anni e si sono sottoposti alla “dieta per la gravidanza”

select count(\*) as numero\_persone

FROM Persona P, EffettuataDa ED, Dieta D

WHERE D.id\_dieta = ED.id\_dieta AND

ED.CF = P.CF AND

DATE\_FORMAT(FROM\_DAYS(DATEDIFF(NOW(), data\_nascita)), '%Y') < 40 AND

D.nome\_dieta = 'dieta per la gravidanza';

1 row in set (0.01 sec)

### Query 10

Si vogliono sapere i nomi dei consorzi che hanno sede legale nella stessa città della sede legale del consorzio “'Agricola Tre Valli s.c.a.”

SELECT C2.nome\_consorzio

FROM Consorzio C1, Consorzio C2

WHERE C1.sede\_legale = C2.sede\_legale AND

C1.nome\_consorzio = 'Agricola Tre Valli s.c.a.' AND

C1.nome\_consorzio <> C2.nome\_consorzio;

2 rows in set (0.00 sec)

### Query 11

Si vuole conoscere, per ogni consorzio, l'area di campo coltivabile totale che hanno a disposizione

SELECT C.nome\_consorzio,sum(T.mq\_terreno)

FROM Consorzio C, Terreno T

WHERE C.partitaIVA = T.partitaIVA

GROUP BY C.nome\_consorzio;

10 rows in set (0.00 sec)

### Query 12

Si vogliono sapere i nomi e i cognomi di tutti gli utenti che hanno almeno una dieta ancora in corso

SELECT DISTINCT P.nome,P.cognome

FROM Persona P,EffettuataDa ED

WHERE P.CF = ED.CF AND

fine\_dieta IS NULL;

2 rows in set (0.00 sec)

### Query 13

Si vuole conoscere, per ogni consorzio, il numero di alimenti distinti che produce

SELECT C.nome\_consorzio, count(DISTINCT id\_prodottoterra) as numero\_alimentiprodotti

FROM Consorzio C, Terreno T, Coltivazione COL

WHERE C.partitaIVA = T.partitaIVA AND

T.id\_terreno = COL.id\_terreno

GROUP BY C.nome\_consorzio;

10 rows in set (0.00 sec)

### Query 14

Si vuole conoscere l’età media delle persone che hanno effettuato la “dieta per allenamento”

SELECT D.nome\_dieta, avg(DISTINCT DATE\_FORMAT(FROM\_DAYS(DATEDIFF(NOW(), P.data\_nascita)), '%Y') ) as età\_media

FROM Dieta D, EffettuataDa ED, Persona P

WHERE D.id\_dieta = ED.id\_dieta AND

ED.CF = P.CF AND

D.nome\_dieta = 'dieta per allenamento';

1 row in set (0.00 sec)

### Query 15

Si vuole conoscere i nomi degli alimenti che sono di stagione tutto l’anno

SELECT PT.nomecomune\_prodottoterra

FROM ProdottoTerra PT, Stagionalità S

WHERE PT.id\_prodottoterra = S.id\_prodottoterra

GROUP BY PT.nomecomune\_prodottoterra

HAVING count(S.nome\_mese) = 12;

7 rows in set (0.01 sec)

### Query 16

Si vuole conoscere il numero di diete iniziate e concluse nel mese di novembre 2022

SELECT count(\*) as numero\_dietedicembre

FROM EffettuataDa ED

WHERE ED.inizio\_dieta BETWEEN '2022-11-01' AND '2022-11-30' AND

ED.fine\_dieta IS NOT NULL;

1 row in set (0.00 sec)

### Query 17

Si vuole sapere qual è l'alimento che possiede meno kcal in 100 grammi di prodotto

SELECT \*

FROM ProdottoTerra PT

ORDER BY PT.kcal\_100g

LIMIT 1;

1 row in set (0.00 sec)

### Query 18

Si vogliono sapere i nomi degli alimenti che stagionano a Novembre con più di 30 milligrammi di Calcio in 100 grammi di prodotto

SELECT PT.nomecomune\_prodottoterra

FROM Mese M, Stagionalità S, ProdottoTerra PT, ContieneSM CSM, SaleMinerale SM

WHERE M.nome\_mese = S.nome\_mese AND

S.id\_prodottoterra = PT.id\_prodottoterra AND

PT.id\_prodottoterra = CSM.id\_prodottoterra AND

CSM.simbolo\_elementochimico = SM.simbolo\_elementochimico AND

SM.nome\_elementochimico = 'Calcio' AND

CSM.quantitàper100g > 0.03 AND

S.nome\_mese = 'novembre';

9 rows in set (0.01 sec)

### Query 19

Si vuole avere la classifica delle prime 10 verdure più coltivate

SELECT PT.nomecomune\_prodottoterra, count(COL.id\_terreno) as numeroterreni\_prodottoterra

FROM ProdottoTerra PT, Coltivazione COL

WHERE PT.id\_prodottoterra = COL.id\_prodottoterra AND

PT.tipo = 'verdura'

GROUP BY PT.nomecomune\_prodottoterra

ORDER BY numeroterreni\_prodottoterra DESC

LIMIT 10;

10 rows in set (0.01 sec)

## Query in Calcolo Relazionale

### Query 3

Si vuole sapere tutta la frutta che è di stagione in estate

{PT.nomecomune\_prodottoterra | PT(ProdottoTerra),S(Stagionalità),M(Mese) | PT.id\_prodottoterra=S.id\_prodottoterra ∧ S.nome\_mese=M.nome\_mese ∧

M.stagione\_mese='estate' ∧ PT.tipo='frutta'}

### Query 18

Si vogliono sapere quali sono gli alimenti che stagionano a Novembre con più di 30 milligrammi di Calcio in 100 grammi di prodotto

{PT.\* | M(Mese),S(Stagionalità),PT(ProdottoTerra),CSM(ContieneSM), SM(SaleMinerale) | M.nome\_mese=S.nome\_mese ∧

S.id\_prodottoterra=PT.id\_prodottoterra ∧

PT.id\_prodottoterra=CSM.id\_prodottoterra ∧

CSM.simbolo\_elementochimico=SM.simbolo\_elementochimico ∧

SM.nome\_elementochimico='Calcio' ∧ CSM.quantitàper100g > 0.03 ∧

S.nome\_mese='novembre'}

## Ottimizzazione Query

In questo caso i tempi di esecuzione sono già molto bassi e le tabelle sono piccole, quindi non si noterebbe il miglioramento; tuttavia, in caso fosse stato necessario, per migliorare i tempi di esecuzione sarebbe opportuno creare indici nelle tabelle, come:

CREATE INDEX IX\_consorzio ON Consorzio(partitaIVA)

CREATE INDEX IX\_prodottoterra ON ProdottoTerra(id\_prodottoterra)

# Trigger, Stored Procedures e Views

## Trigger

I trigger, insieme ai check, vengono usati per mantenere la validità dei dati all’interno della base di dati. Di seguito riporto alcuni trigger utili al database:

### Before\_insert\_anno\_acquistoterreno

Questo trigger verifica che ogni volta che viene inserito un nuovo terreno nella tabella Terreno, l’anno d’acquisto di questo sia posteriore all’anno di fondazione del consorzio a cui appartiene e, in caso contrario, lo imposta al primo anno di esistenza di quel consorzio. Inoltre, imposta al minimo il numero dei dipendenti e la superficie del terreno nel caso vengano inseriti valori invalidi.

DELIMITER |

CREATE TRIGGER before\_insert\_anno\_acquistoterreno

BEFORE INSERT ON Terreno FOR EACH ROW

BEGIN

DECLARE annofondazione YEAR;

SELECT anno\_fondazione

FROM Consorzio

WHERE partitaIVA = NEW.partitaIVA

INTO annofondazione;

IF NEW.anno\_acquistoterreno < annofondazione THEN SET NEW.anno\_acquistoterreno = annofondazione;

END IF;

IF NEW.numero\_dipendenti < 0 THEN SET NEW.numero\_dipendenti = 0;

END IF;

IF NEW.mq\_terreno <= 0 THEN SET NEW.mq\_terreno = 1;

END IF;

END;

|

### Before\_update\_anno\_acquistoterreno

Simile al precedente ma per gli update

DELIMITER |

CREATE TRIGGER before\_update\_anno\_acquistoterreno

BEFORE UPDATE ON Terreno FOR EACH ROW

BEGIN

DECLARE annofondazione YEAR;

SELECT anno\_fondazione

FROM Consorzio

WHERE partitaIVA = NEW.partitaIVA

INTO annofondazione;

IF NEW.anno\_acquistoterreno < annofondazione THEN SET NEW.anno\_acquistoterreno = annofondazione;

END IF;

IF NEW.numero\_dipendenti < 0 THEN SET NEW.numero\_dipendenti = 0;

END IF;

IF NEW.mq\_terreno <= 0 THEN SET NEW.mq\_terreno = 1;

END IF;

END;

|

### Before\_insert\_durata\_diete

Questo trigger verifica che la data di fine della dieta (se impostata) sia uguale o posteriore alla data d’inizio.

DELIMITER |

CREATE TRIGGER before\_insert\_durata\_diete

BEFORE INSERT ON EffettuataDa FOR EACH ROW

BEGIN

IF NEW.inizio\_dieta > NEW.fine\_dieta THEN SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'ERRORE : date non coerenti';

END IF;

END;

|

### Before\_update\_durata\_diete

Simile al precedente ma per gli update

DELIMITER |

CREATE TRIGGER before\_update\_durata\_diete

BEFORE UPDATE ON EffettuataDa FOR EACH ROW

BEGIN

IF NEW.inizio\_dieta > NEW.fine\_dieta THEN SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'ERRORE : date non coerenti';

END IF;

END;

|

## Stored Procedures

Di seguito riporto alcune procedures utili agli utenti che dovranno usufruire del database:

### Nuovo\_utente

Questa procedure consente ai nuovi utenti di iscriversi al database inserendo le proprie informazioni personali (dato che gli utenti non hanno i privilegi di modificare le tabelle)

DELIMITER |

CREATE PROCEDURE nuovo\_utente(in nuCF CHAR(16), in nunome VARCHAR(50), in nucognome VARCHAR(50), in nudata\_nascita DATE, in nuindirizzo VARCHAR(100), in nutelefono VARCHAR(13), in nuemail VARCHAR(50))

BEGIN

INSERT INTO Persona (CF, nome, cognome, data\_nascita, indirizzo, telefono, email)

VALUES (nuCF,nunome,nucognome,nudata\_nascita,nuindirizzo,nutelefono,nuemail);

END;

|

### Nuova\_dieta

Questa procedure consente agli utenti già registrati di inserire nel database la dieta che vogliono iniziare tra quelle già presenti inserendo il loro codice fiscale, il nome della dieta, una data d’inizio e un’eventuale data di fine.

DELIMITER |

CREATE PROCEDURE nuova\_dieta(in ndCF CHAR(16), in ndnome\_dieta VARCHAR(50), in nddata\_inizio DATE, in nddata\_fine DATE)

BEGIN

DECLARE ndid\_dieta VARCHAR(10);

SELECT id\_dieta

FROM Dieta

WHERE nome\_dieta = ndnome\_dieta

INTO ndid\_dieta;

INSERT INTO EffettuataDa(CF,id\_dieta,inizio\_dieta,fine\_dieta)

VALUES(ndCF,ndid\_dieta,nddata\_inizio,nddata\_fine);

END;

|

### Inserisci\_fine\_dieta

Questa procedure consente agli utenti di impostare o modificare la data di fine di una dieta già inserita nel database inserendo, oltre alla data di fine: codice fiscale, nome della dieta e data in cui aveva avuto inizio.

DELIMITER |

CREATE PROCEDURE inserisci\_fine\_dieta(in ifdCF CHAR(16), in ifdnome\_dieta VARCHAR(50), in ifddata\_inizio DATE, in ifddata\_fine DATE)

BEGIN

DECLARE ifdid\_dieta VARCHAR(10);

SELECT id\_dieta

FROM Dieta

WHERE nome\_dieta = ifdnome\_dieta

INTO ifdid\_dieta;

UPDATE EffettuataDa

SET fine\_dieta = ifddata\_fine

WHERE CF = ifdCF AND id\_dieta = ifdid\_dieta AND inizio\_dieta = ifddata\_inizio;

END;

|

## Views

Di seguito alcune views per rendere permanenti alcune delle interrogazioni più frequenti. Potranno anche esser usate come punto di partenza per nuove interrogazioni.

### ProdottiStagionali\_PerMese

Mantiene, per ogni mese, i prodotti di stagione

CREATE VIEW ProdottiStagionali\_PerMese AS

SELECT M.nome\_mese as mese,PT.nomecomune\_prodottoterra as prodotto\_terra

FROM ProdottoTerra PT,Stagionalità S,Mese M

WHERE PT.id\_prodottoterra = S.id\_prodottoterra AND

S.nome\_mese = M.nome\_mese

ORDER BY M.nome\_mese;

### Alimenti\_PerDieta

Mantiene, per ogni dieta, gli alimenti consigliati per soddisfare il fabbisogno giornaliero richiesto dalla dieta.

CREATE VIEW Alimenti\_PerDieta AS

SELECT D.nome\_dieta,PT.nomecomune\_prodottoterra

FROM Dieta D,VUsataIn VIN,Vitamina V,ContieneV CV,ProdottoTerra PT

WHERE D.id\_dieta = VIN.id\_dieta AND

VIN.nome\_comunevitamina = V.nome\_comunevitamina AND

V.nome\_comunevitamina = CV.nome\_comunevitamina AND

CV.id\_prodottoterra = PT.id\_prodottoterra

UNION

SELECT D2.nome\_dieta,PT2.nomecomune\_prodottoterra

FROM Dieta D2, SMUsatoIn SMIN,SaleMinerale SM,ContieneSM CSM,ProdottoTerra PT2

WHERE D2.id\_dieta = SMIN.id\_dieta AND

SMIN.simbolo\_elementochimico = SM.simbolo\_elementochimico AND

SM.simbolo\_elementochimico = CSM.simbolo\_elementochimico AND

CSM.id\_prodottoterra = PT2.id\_prodottoterra

ORDER BY nome\_dieta;

### Funzioni\_Vitamine

Mantiene, per ogni vitamina, le funzioni che svolge all’interno del nostro corpo

CREATE VIEW Funzioni\_Vitamine AS

SELECT V.nome\_comunevitamina, F.nome\_funzione,F.descrizione

FROM Vitamina V, VSvolge VS, Funzione F

WHERE V.nome\_comunevitamina = VS.nome\_comunevitamina AND

VS.id\_funzione = F.id\_funzione

ORDER BY V.nome\_comunevitamina ASC;

### Funzioni\_SaliMinerali

Mantiene, per ogni sale minerale, le funzioni che svolge all’interno del nostro corpo

CREATE VIEW Funzioni\_SaliMinerali AS

SELECT SM.nome\_elementochimico, F.nome\_funzione,F.descrizione

FROM SaleMinerale SM, SMSvolge SMS, Funzione F

WHERE SM.simbolo\_elementochimico = SMS.simbolo\_elementochimico AND

SMS.id\_funzione = F.id\_funzione

ORDER BY SM.nome\_elementochimico ASC;

# Sicurezza

In questo database ci sono due tipologie di utenti:

* **Admin** : amministratore dell’intero database
* **User** : utenti che usufruiscono delle funzionalità del Database

L’Admin possiede i permessi di lettura, inserimento, modifica e cancellazione su tutte le tabelle, mentre gli User hanno varie restrizioni, tra le quali:

* Non possono modificare, inserire o cancellare dati in nessuna tabella
* Possono effettuare operazioni di lettura su ogni tabella eccetto Persona e EffettuataDa
* Possono usare le Stored Procedures (ma usando le informazioni del proprio account e non di altri)

I permessi possono esser forniti e revocati attraverso i comandi di **Grant** e **Revoke**. Degli esempi possono essere:

**GRANT all privileges on \*.\* to 'root’@’localhost’ identified by 'Password1’ with grant option;**

**GRANT select on Dieta to 'user\_a'@’stagionalitacibi.example.com’ identified by 'Password1’;**

# INSERT

Di seguito il codice di alcuni inserimenti opportuni per ciascuna tabella.

## Mese

INSERT INTO Mese (nome\_mese, stagione\_mese) VALUES ('gennaio', 'inverno'),

('febbraio', 'inverno'),

('marzo', 'primavera'),

('aprile', 'primavera'),

('maggio', 'primavera'),

('giugno', 'estate'),

('luglio', 'estate'),

('agosto', 'estate'),

('settembre', 'autunno'),

('ottobre', 'autunno'),

('novembre', 'autunno'),

('dicembre', 'inverno');

## Vitamina

INSERT INTO Vitamina(nome\_comunevitamina, nome\_chimicovitamina, tipo)

VALUES ('A', 'retinolo', 'liposolubile'),

('B1', 'tiamina', 'idrosolubile'),

('B2', 'riboflavina', 'idrosolubile'),

('B3', 'niacina', 'idrosolubile'),

('B5', 'acido pantotenico', 'idrosolubile'),

('B6', 'piridossina', 'idrosolubile'),

('B8', 'biotina', 'idrosolubile'),

('B9', 'acido folico', 'idrosolubile'),

('C', 'acido ascorbico', 'idrosolubile'),

('D', 'calciferolo', 'liposolubile'),

('E', 'tocoferolo', 'liposolubile'),

('K', 'menadione', 'liposolubile');

## SaleMinerale

INSERT INTO SaleMinerale(simbolo\_elementochimico, nome\_elementochimico, tipo)

VALUES ('Ca', 'calcio', 'macroelemento'),

('P', 'fosforo', 'macroelemento'),

('Na', 'sodio', 'macroelemento'),

('K', 'potassio', 'macroelemento'),

('Cl', 'cloro', 'macroelemento'),

('Mg', 'magnesio', 'macroelemento'),

('Fe', 'ferro', 'microelemento');

INSERT INTO Persona (CF, nome, cognome, data\_nascita, indirizzo, telefono, email)

VALUES ('MNNGRM90P17H501D', 'girolamo', 'manno', '1990-11-23', 'via amatrice 2', '3123456781', 'girolamo.manno@gmail.com'),

('BRTMRM90P16H501D', 'miriam', 'bertolini', '1989-04-15', 'via roma 9', '3123456782', 'miriam.bertolini@gmail.com'),

('HDZANT80P18H501D', 'anita', 'hadzhiivanova', '1980-09-09', 'via nettunense 12', '3123456783', 'anita.hadzhiivanova@gmail.com'),

('HDZIVO68P18H501D', 'ivo', 'hadzhiivanov', '1968-10-17', 'via g.garibaldi 34', '3763566777', 'ivo.hadzhiivanov@gmail.com'),

('CMNRSL99P11H501D', 'rosella', 'cimini', '1999-01-12', 'via umberto I 28', '3123456785', 'rosella.cimini@gmail.com'),

('CSTDVD75P19H501D', 'davide', 'costa', '1975-08-22', 'via bari 11', '3123456786', 'davide.costa@gmail.com'),

('CTSDMN66P14H501D', 'damiano', 'catasta', '1966-07-19', 'via napoli 56', '3123456787', 'damiano.catasta@libero.it'),

('XUUFRC84P22H501D', 'francesco', 'xu', '1984-11-05', 'viale kennedy 101', '3123456788', 'francesco.xu@alice.it'),

('CMNGLA91P11H501D', 'giulia', 'cimini', '1991-12-12', 'via M.L.King 87', '3123456789', 'giulia.cimini@libero.it'),

('PLCNCL79P10H501D', 'nicolas', 'palacio', '1979-03-16', 'via milano 66', '3123456790', 'nicolas.palacio@hotmail.it'),

('NVCSRA99P17H501D', 'sara', 'novac', '1999-01-12', 'via degli alberini 33', '3123456791', 'sara.novac@gmail.com'),

('PRSRTA40P12H501D', 'rita', 'persichilli', '1940-04-29', 'via degli umbri 16', '3123667360', NULL),

('ATNFDR99P11H501D', 'federica', 'antonucci', '1999-07-16', 'via della chiesa 45', '312256790', 'federica.antonucci@libero.it'),

('MRCANN98P21H501D', 'anna', 'marcaccini', '1998-06-03', 'via colle sant\'agostino 11', '3336557790', 'anna.marcaccini@hotmail.it'),

('ACCMRI79P11H501D', 'maria', 'accietto', '1979-01-26', 'via francesco borsellino 23', '3343456590', 'maria.accietto@hotmail.it');

INSERT INTO ProdottoTerra (id\_prodottoterra,nomecomune\_prodottoterra,nomescientifico\_prodottoterra,kcal\_100g,tipo)

VALUES ('C1', 'aglio', 'allium sativum', 53, 'verdura'),

('C2', 'albicocche', 'prunus armeniaca', 42, 'frutta'),

('C3', 'ananas', 'ananas sativa', 42, 'frutta'),

('C4', 'anguria','citrusllus vulgaris', 16, 'frutta'),

('C5', 'arance','citrus sinensis', 37, 'frutta'),

('C6', 'banana', 'musa sapientium', 76, 'frutta'),

('C7', 'bieta', 'bieta vulgaris', 15, 'verdura'),

('C8', 'broccoli', 'brassica oleracea', 33, 'verdura'),

('C9', 'carciofi','cynara scolymus', 33, 'verdura'),

('C10', 'carote', 'daucus carota', 41, 'verdura'),

('C11', 'castagne', 'castanea vulgaris', 174, 'frutta'),

('C12', 'cavolfiori', 'brassica oleracea', 30, 'verdura'),

('C13', 'cetrioli', 'cucumis sativus', 16,'verdura'),

('C14', 'ciliegie', 'prunus avium', 48, 'frutta'),

('C15', 'cipolle', 'allium cepa', 28,'verdura'),

('C16', 'fichi', 'ficus carica', 63, 'frutta'),

('C17', 'finocchi', 'foeniculum vulgare', 15, 'verdura'),

('C18', 'fragole', 'fragaria vesca', 30, 'frutta'),

('C19', 'kiwi', 'actinidia deliciosa', 48, 'frutta'),

('C20', 'limoni', 'citrus limon', 16, 'frutta'),

('C21', 'mandarini', 'citrus aurantium', 76, 'frutta'),

('C22', 'melanzane', 'solanum melongena', 23, 'verdura'),

('C23', 'mele', 'pyrus malus', 44, 'frutta'),

('C24', 'melone', 'cucumis melo', 34, 'frutta'),

('C25', 'mirtilli', 'vaccinium myrtillus', 49, 'frutta'),

('C26', 'patate', 'solanum tuberosum', 72, 'verdura'),

('C27', 'peperoni', 'caspicum annuum', 26, 'verdura'),

('C28', 'pere', 'pyrus communis', 43, 'frutta'),

('C29', 'pesche', 'prunus persica', 28, 'frutta'),

('C30', 'piselli', 'pisum sativum', 64, 'verdura'),

('C31', 'pomodori', 'solanum lycopersicum', 23, 'verdura'),

('C32', 'pompelmo', 'cytrus paradisi', 29,'frutta'),

('C33', 'prugne', 'prunus domestica', 45, 'frutta'),

('C34', 'radicchio', 'cichorium intybus', 19, 'verdura'),

('C35', 'rape', 'brassica rapa', 24, 'verdura'),

('C36', 'rucola', 'eruca sativa', 30, 'verdura'),

('C37', 'spinaci', 'spinacia oleracea', 35, 'verdura'),

('C38', 'uva', 'vitis vinifera', 64, 'frutta'),

('C39', 'zucca', 'cucurbita maxima', 29, 'verdura'),

('C40', 'zucchine', 'cucurbita pepo', 16,'verdura');

## Stagionalità

INSERT INTO Stagionalità(nome\_mese,id\_prodottoterra)

VALUES ('gennaio','C5'),

('gennaio', 'C21'),

('gennaio', 'C23'),

('gennaio', 'C28'),

('gennaio', 'C32'),

('gennaio', 'C7'),

('gennaio', 'C8'),

('gennaio', 'C12'),

('gennaio', 'C17'),

('gennaio', 'C34'),

('gennaio', 'C35'),

('gennaio', 'C39'),

('febbraio', 'C5'),

('febbraio', 'C21'),

('febbraio', 'C7'),

('febbraio', 'C23'),

('febbraio', 'C28'),

('febbraio', 'C32'),

('febbraio', 'C8'),

('febbraio', 'C9'),

('febbraio', 'C12'),

('febbraio', 'C17'),

('febbraio', 'C34'),

('febbraio', 'C35'),

('febbraio', 'C39'),

('marzo', 'C5'),

('marzo', 'C23'),

('marzo', 'C28'),

('marzo', 'C32'),

('marzo', 'C7'),

('marzo', 'C8'),

('marzo', 'C9'),

('marzo', 'C12'),

('marzo', 'C17'),

('marzo', 'C37'),

('marzo', 'C34'),

('marzo', 'C35'),

('marzo', 'C39'),

('aprile', 'C5'),

('aprile', 'C18'),

('aprile', 'C23'),

('aprile', 'C28'),

('aprile', 'C7'),

('aprile', 'C9'),

('aprile', 'C12'),

('aprile', 'C17'),

('aprile', 'C30'),

('aprile', 'C37'),

('maggio', 'C2'),

('maggio', 'C14'),

('maggio', 'C18'),

('maggio', 'C23'),

('maggio', 'C28'),

('maggio', 'C9'),

('maggio', 'C13'),

('maggio', 'C30'),

('maggio', 'C37'),

('maggio', 'C40'),

('giugno', 'C2'),

('giugno', 'C4'),

('giugno', 'C14'),

('giugno', 'C25'),

('giugno', 'C29'),

('giugno', 'C33'),

('giugno', 'C13'),

('giugno', 'C22'),

('giugno', 'C27'),

('giugno', 'C30'),

('giugno', 'C31'),

('giugno', 'C37'),

('giugno', 'C40'),

('luglio', 'C2'),

('luglio', 'C4'),

('luglio', 'C16'),

('luglio', 'C24'),

('luglio', 'C25'),

('luglio', 'C29'),

('luglio', 'C33'),

('luglio', 'C13'),

('luglio', 'C30'),

('luglio', 'C27'),

('luglio', 'C31'),

('luglio', 'C37'),

('luglio', 'C40'),

('agosto', 'C2'),

('agosto', 'C4'),

('agosto', 'C16'),

('agosto', 'C24'),

('agosto', 'C25'),

('agosto', 'C29'),

('agosto', 'C33'),

('agosto', 'C38'),

('agosto', 'C13'),

('agosto', 'C22'),

('agosto', 'C27'),

('agosto', 'C31'),

('agosto', 'C37'),

('agosto', 'C39'),

('agosto', 'C40'),

('settembre', 'C11'),

('settembre', 'C16'),

('settembre', 'C19'),

('settembre', 'C23'),

('settembre', 'C24'),

('settembre', 'C29'),

('settembre', 'C33'),

('settembre', 'C38'),

('settembre', 'C13'),

('settembre', 'C22'),

('settembre', 'C27'),

('settembre', 'C31'),

('settembre', 'C34'),

('settembre', 'C37'),

('settembre', 'C39'),

('settembre', 'C40'),

('ottobre', 'C5'),

('ottobre', 'C11'),

('ottobre', 'C19'),

('ottobre', 'C23'),

('ottobre', 'C7'),

('ottobre', 'C38'),

('ottobre', 'C8'),

('ottobre', 'C12'),

('ottobre', 'C17'),

('ottobre', 'C34'),

('ottobre', 'C40'),

('novembre', 'C5'),

('novembre', 'C11'),

('novembre', 'C19'),

('novembre', 'C21'),

('novembre', 'C23'),

('novembre', 'C28'),

('novembre', 'C7'),

('novembre', 'C32'),

('novembre', 'C8'),

('novembre', 'C17'),

('novembre', 'C34'),

('novembre', 'C35'),

('novembre', 'C39'),

('dicembre', 'C5'),

('dicembre', 'C19'),

('dicembre', 'C21'),

('dicembre', 'C23'),

('dicembre', 'C28'),

('dicembre', 'C32'),

('dicembre', 'C7'),

('dicembre', 'C8'),

('dicembre', 'C17'),

('dicembre', 'C34'),

('dicembre', 'C35'),

('dicembre', 'C39'),

('gennaio', 'C3'),

('febbraio', 'C3'),

('marzo', 'C3'),

('aprile', 'C3'),

('maggio', 'C3'),

('giugno', 'C3'),

('luglio', 'C3'),

('agosto', 'C3'),

('settembre', 'C3'),

('ottobre', 'C3'),

('novembre', 'C3'),

('dicembre', 'C3'),

('gennaio', 'C6'),

('febbraio', 'C6'),

('marzo', 'C6'),

('aprile', 'C6'),

('maggio', 'C6'),

('giugno', 'C6'),

('luglio', 'C6'),

('agosto', 'C6'),

('settembre', 'C6'),

('ottobre', 'C6'),

('novembre', 'C6'),

('dicembre', 'C6'),

('gennaio', 'C1'),

('febbraio', 'C1'),

('marzo', 'C1'),

('aprile', 'C1'),

('maggio', 'C1'),

('giugno', 'C1'),

('luglio', 'C1'),

('agosto', 'C1'),

('settembre', 'C1'),

('ottobre', 'C1'),

('novembre', 'C1'),

('dicembre', 'C1'),

('gennaio', 'C10'),

('febbraio', 'C10'),

('marzo', 'C10'),

('aprile', 'C10'),

('maggio', 'C10'),

('giugno', 'C10'),

('luglio', 'C10'),

('agosto', 'C10'),

('settembre', 'C10'),

('ottobre', 'C10'),

('novembre', 'C10'),

('dicembre', 'C10'),

('gennaio', 'C15'),

('febbraio', 'C15'),

('marzo', 'C15'),

('aprile', 'C15'),

('maggio', 'C15'),

('giugno', 'C15'),

('luglio', 'C15'),

('agosto', 'C15'),

('settembre', 'C15'),

('ottobre', 'C15'),

('novembre', 'C15'),

('dicembre', 'C15'),

('gennaio', 'C26'),

('febbraio', 'C26'),

('marzo', 'C26'),

('aprile', 'C26'),

('maggio', 'C26'),

('giugno', 'C26'),

('luglio', 'C26'),

('agosto', 'C26'),

('settembre', 'C26'),

('ottobre', 'C26'),

('novembre', 'C26'),

('dicembre', 'C26'),

('gennaio', 'C36'),

('febbraio', 'C36'),

('marzo', 'C36'),

('aprile', 'C36'),

('maggio', 'C36'),

('giugno', 'C36'),

('luglio', 'C36'),

('agosto', 'C36'),

('settembre', 'C36'),

('ottobre', 'C36'),

('novembre', 'C36'),

('dicembre', 'C36');

## ContieneSM

INSERT INTO ContieneSM(id\_prodottoterra,simbolo\_elementochimico,quantitàper100g)

VALUES ('C1','Na', 0.004),

('C1','K',0.31),

('C1','Ca',0.015),

('C1','Mg',0.017),

('C1','P',0.106),

('C1','Fe',0.0012),

('C2','Na',0.001),

('C2','K',0.32),

('C2','Ca',0.016),

('C2','P',0.016),

('C2','Fe',0.0005),

('C3','Na',0.002),

('C3','K',0.25),

('C3','Ca',0.017),

('C3','P',0.008),

('C3','Fe',0.0005),

('C4','Na',0.003),

('C4','K',0.28),

('C4','Ca',0.007),

('C4','P',0.002),

('C4','Fe',0.0002),

('C5','Na',0.003),

('C5','K',0.2),

('C5','Ca',0.049),

('C5','P',0.022),

('C5','Fe',0.0002),

('C6','Na',0.001),

('C6','K',0.35),

('C6','Ca',0.007),

('C6','P',0.028),

('C6','Fe',0.0008),

('C7','Na',0.01),

('C7','K',0.196),

('C7','Ca',0.067),

('C7','Mg',0.038),

('C7','P',0.029),

('C7','Fe',0.001),

('C8','Na',0.012),

('C8','K',0.34),

('C8','Ca',0.28),

('C8','P',0.066),

('C8','Fe',0.0008),

('C9','Na',0.133),

('C9','K',0.376),

('C9','Ca',0.086),

('C9','Mg',0.045),

('C9','P',0.067),

('C9','Fe',0.001),

('C10','Na', 0.095),

('C10','K',0.22),

('C10','Ca',0.044),

('C10','Mg',0.11),

('C10','P',0.037),

('C10','Fe',0.0007),

('C11','Na',0.009),

('C11','K',0.395),

('C11','Ca',0.03),

('C11','P',0.081),

('C11','Fe',0.0009),

('C12','Na', 0.008),

('C12','K', 0.35),

('C12','Ca', 0.044),

('C12','Mg', 0.028),

('C12','P', 0.069),

('C12','Fe',0.0008),

('C13','Na', 0.013),

('C13','K',0.14),

('C13','Ca',0.016),

('C13','P',0.017),

('C13','Fe',0.0003),

('C14','Na',0.003),

('C14','K',0.229),

('C14','Ca',0.03),

('C14','P',0.018),

('C14','Fe',0.0006),

('C15','Na', 0.01),

('C15','K', 0.14),

('C15','Ca', 0.025),

('C15','P', 0.035),

('C15','Fe', 0.0004),

('C16','Na',0.002),

('C16','K',0.27),

('C16','Ca',0.043),

('C16','P',0.025),

('C16','Fe',0.0005),

('C17','Na',0.004),

('C17','K',0.394),

('C17','Ca',0.045),

('C17','Mg',0.016),

('C17','P',0.039),

('C17','Fe',0.0004),

('C18','Na',0.002),

('C18','K', 0.16),

('C18','Ca',0.035),

('C18','P',0.028),

('C18','Fe',0.0008),

('C19','Na',0.005),

('C19','K',0.4),

('C19','Ca',0.025),

('C19','Mg',0.012),

('C19','P',0.07),

('C19','Fe',0.0005),

('C20','Na',0.002),

('C20','K',0.14),

('C20','Ca',0.014),

('C20','P',0.011),

('C20','Fe',0.0001),

('C21','Na',0.001),

('C21','K',0.21),

('C21','Ca',0.032),

('C21','P',0.019),

('C21','Fe',0.0003),

('C22','Na',0.003),

('C22','K',0.19),

('C22','Ca',0.015),

('C22','Mg',0.013),

('C22','P',0.02),

('C22','Fe',0.002),

('C23','Na', 0.002),

('C23','K', 0.076),

('C23','Ca', 0.002),

('C23','P', 0.006),

('C23','Fe', 0.0001),

('C24','Na',0.008),

('C24','K',0.333),

('C24','Ca',0.019),

('C24','P',0.013),

('C24','Fe',0.0003),

('C25','Na',0.002),

('C25','K',0.16),

('C25','Ca',0.041),

('C25','P',0.031),

('C25','Fe',0.0007),

('C26','Na',0.007),

('C26','K',0.57),

('C26','Ca',0.01),

('C26','Mg',0.028),

('C26','P',0.054),

('C26','Fe',0.0006),

('C27','Na',0.002),

('C27','K',0.21),

('C27','Ca',0.017),

('C27','P',0.028),

('C27','Fe',0.0007),

('C28','Na',0.002),

('C28','K',0.127),

('C28','Ca',0.011),

('C28','Mg',0.007),

('C28','P',0.015),

('C28','Fe',0.0003),

('C29','Na',0.003),

('C29','K',0.26),

('C29','Ca',0.008),

('C29','Mg',0.009),

('C29','P',0.02),

('C29','Fe',0.0004),

('C30','Na',0.001),

('C30','K',0.193),

('C30','Ca',0.045),

('C30','P',0.097),

('C30','Fe',0.0017),

('C31','Na',0.006),

('C31','K',0.297),

('C31','Ca',0.009),

('C31','Mg',0.01),

('C31','P',0.025),

('C31','Fe',0.0003),

('C32','Na',0.001),

('C32','K',0.23),

('C32','Ca',0.017),

('C32','P',0.016),

('C32','Fe',0.0003),

('C33','Na',0.002),

('C33','K',0.19),

('C33','Ca',0.013),

('C33','P',0.014),

('C33','Fe',0.0002),

('C34','Na',0.01),

('C34','K',0.24),

('C34','Ca',0.036),

('C34','P',0.03),

('C34','Fe',0.0003),

('C35','Na',0.015),

('C35','K',0.28),

('C35','Ca',0.04),

('C35','P',0.029),

('C35','Fe',0.0006),

('C36','K',0.468),

('C36','Ca',0.309),

('C36','P',0.041),

('C36','Fe',0.0052),

('C37','Na',0.1),

('C37','K',0.53),

('C37','Ca',0.078),

('C37','Mg',0.06),

('C37','P',0.062),

('C37','Fe',0.0029),

('C38','Na',0.001),

('C38','K',0.192),

('C38','Ca',0.033),

('C38','Mg',0.007),

('C38','P',0.022),

('C38','Fe',0.0004),

('C39','Ca',0.02),

('C39','P',0.04),

('C39','Fe',0.0009),

('C40','Na',0.001),

('C40','K',0.29),

('C40','Ca',0.018),

('C40','Mg',0.022),

('C40','P',0.055),

('C40','Fe',0.0005);

## ContieneV

INSERT INTO ContieneV(id\_prodottoterra,nome\_comunevitamina,quantitàper100g)

VALUES ('C1','B1',0.00023),

('C1','B2',0.00007),

('C1','B3',0.00073),

('C1','B6',0.00129),

('C1','C',0.009),

('C2','B1',0.00003),

('C2','B2',0.00003),

('C2','B3',0.0005),

('C2','A',0.00036),

('C3','B1',0.00005),

('C3','B2',0.00001),

('C3','B3',0.0002),

('C3','C',0.017),

('C3','A',0.007),

('C4','B1',0.00002),

('C4','B2',0.00002),

('C4','C',0.008),

('C4','A',0.037),

('C5','B1',0.00006),

('C5','B2',0.00005),

('C5','B3',0.0002),

('C5','C',0.05),

('C5','D',0.000020),

('C5','A',0.000071),

('C6','B1',0.00006),

('C6','B2',0.00006),

('C6','B3',0.0007),

('C6','C',0.016),

('C6','A',0.000045),

('C7','B1',0.00003),

('C7','B2',0.00019),

('C7','B3',0.0018),

('C7','C',0.024),

('C7','B9',0.000085),

('C7','A',0.000263),

('C7','E',0.005),

('C8','B1',0.00004),

('C8','B2',0.00012),

('C8','B3',0.0018),

('C8','C',0.054),

('C8','B9',0.000125),

('C9','B1',0.00006),

('C9','B2',0.0001),

('C9','B3',0.0005),

('C9','C',0.012),

('C9','A',0.000018),

('C10','C',0.004),

('C10','D',0.000015),

('C10','A',0.001148),

('C11','B3',0.001),

('C12','B3',0.0012),

('C12','C', 0.059),

('C12','B9', 0.000071),

('C12','A', 0.00005),

('C13','C',0.011),

('C14','B1',0.00003),

('C14','B2',0.00003),

('C14','B3',0.0005),

('C14','C',0.011),

('C14','B9',0.000007),

('C14','A',0.000019),

('C15','B1',0.00003),

('C15','B2',0.00003),

('C15','B3',0.0005),

('C15','C',0.005),

('C15','B9',0.00004),

('C15','A', 0.000003),

('C16','B1',0.00003),

('C16','B2',0.00004),

('C16','B3',0.0004),

('C16','C',0.007),

('C16','A',0.000015),

('C17','B1',0.00002),

('C17','B2',0.00004),

('C17','B3',0.0005),

('C17','C',0.012),

('C17','A',0.000002),

('C18','B1',0.00001),

('C18','B2',0.00004),

('C18','B3',0.0005),

('C18','C',0.054),

('C19','B1',0.00002),

('C19','B2',0.00005),

('C19','B3',0.0004),

('C19','C',0.085),

('C20','B1',0.00004),

('C20','B2',0.00001),

('C20','B3',0.0003),

('C20','C',0.05),

('C21','B1',0.00008),

('C21','B2',0.00007),

('C21','B3',0.0003),

('C21','C',0.042),

('C21','B9',0.000038),

('C21','A',0.000018),

('C22','B1',0.00005),

('C22','B2',0.00005),

('C22','B3',0.0006),

('C22','C',0.011),

('C22','B9',0.02),

('C23','B1',0.00021),

('C23','B2',0.00002),

('C23','B3',0.00001),

('C23','C',0.003),

('C24','B1',0.00005),

('C24','B2',0.00004),

('C24','B3',0.0006),

('C24','C',0.032),

('C24','A',0.189),

('C25','B1',0.00002),

('C25','B2',0.00005),

('C25','B3',0.0005),

('C25','C',0.015),

('C25','A',0.013),

('C26','B1',0.0001),

('C26','B2',0.00004),

('C26','B3',0.0025),

('C26','C',0.016),

('C26','B9',0.000038),

('C26','A',0.000003),

('C27','B1',0.00005),

('C27','B2',0.00007),

('C27','B3',0.0005),

('C27','C',0.151),

('C27','A',0.000139),

('C28','B1',0.00001),

('C28','B2',0.00003),

('C28','B3',0.0001),

('C28','C',0.004),

('C28','B9',0.000005),

('C29','B1',0.00001),

('C29','B2',0.00003),

('C29','B3',0.0005),

('C29','C',0.004),

('C30','B1',0.0002),

('C30','B2',0.00008),

('C30','B3',0.0026),

('C30','C',0.032),

('C30','A',0.064),

('C31','B1',0.00002),

('C31','B2',0.0008),

('C31','B6',0.00013),

('C31','C',0.025),

('C31','A',0.00061),

('C32','B1',0.00005),

('C32','B2',0.00003),

('C32','B3',0.0002),

('C32','C',0.04),

('C32','B9',0.027),

('C33','B1',0.00008),

('C33','B2',0.00005),

('C33','B3',0.0005),

('C33','C',0.005),

('C33','A',0.000016),

('C33','K',0.0000117),

('C34','B1',0.00007),

('C34','B2',0.00005),

('C34','B3',0.0003),

('C34','B6',0.00022),

('C34','C',0.01),

('C35','B1',0.00002),

('C35','B2',0.00007),

('C35','B3',0.0009),

('C35','C',0.023),

('C36','C',0.11),

('C36','B9',0.099),

('C36','A',0.000742),

('C37','B1',0.00007),

('C37','B2',0.00037),

('C37','B3',0.0004),

('C37','C',0.054),

('C37','A',0.000485),

('C38','B1',0.00003),

('C38','B2',0.00003),

('C38','B3',0.0004),

('C38','C',0.006),

('C38','A',0.000018),

('C38','E',0.0001),

('C39','B1',0.00003),

('C39','B2',0.00002),

('C39','B3',0.0005),

('C39','C',0.009),

('C39','A',0.000599),

('C40','B1',0.00008),

('C40','B2',0.00012),

('C40','B3',0.0007),

('C40','C',0.011),

('C40','A',0.000006);

## Consorzio

INSERT INTO Consorzio(partitaIVA, nome\_consorzio, sede\_legale, anno\_fondazione, telefono)

VALUES (101283360, 'Agrofruit s.r.l.', 'Brescia', '1965', '331283360'),

(101285828, 'Ortoro s.r.l.', 'Bari', '1999', '331285828'),

(101284594, 'Valfrutta', 'Bologna', '1960', '331284594'),

(101288296, 'A.a. Feresin', 'Bolzano', '1960', '331288296'),

(101289530, 'Distribuzione s.r.l.', 'Bolzano', '1977', '331289530'),

(101290764, 'Raccolto sano s.r.l.', 'Roma', '1974', '331290764'),

(101287062, 'Verdura Valle D\'Oro soc. coop.','Trento', '1946', '331287062'),

(101291998, 'Ortofrutta s.r.l.', 'Roma', '2001', '331291998'),

(101293232, 'La Riserva Bio s.r.l', 'Napoli', '2005', '331293232'),

(101294466, 'Agricola Tre Valli s.c.a.', 'Bolzano', '1987', '331294466');

## Terreno

INSERT INTO Terreno(id\_terreno,indirizzo,mq\_terreno,anno\_acquistoterreno,numero\_dipendenti,partitaIVA)

VALUES ('T1','Borgo Colombo 540' ,17000,'2000',70,101285828),

('T2','Via Riviera Mallozzi 25',10000,'1966',50,101283360),

('T3','Via Monte Grappa 33',15000,'2002',60,101285828),

('T4','Via delle Mura Castellane 56',20000,'1970',100,101284594),

('T5','Viale Antonio Vivaldi 36',17000,'1972',70,101288296),

('T6','Via Lombardi 59',12000,'1980',65,101289530),

('T7','Via Aldobrandini 9',11000,'1975',40,101290764),

('T8','Via Morlupo 21',16000,'1949',70,101287062),

('T9','Via dello Scafo 392',13500,'2002',85,101291998),

('T10','Via Corio 45',25000,'2007',110,101293232),

('T11','Via Bonafous Alfonso 127',15000,'1988',80,101294466),

('T12','Via Pasquale Scura 78',17000,'2005',55,101285828),

('T13','Via Santa Lucia 48',10000,'1982',30,101284594),

('T14','Via delle Coste 135',12000,'1987',50,101284594),

('T15','Via Piave 38',14000,'1998',60,101290764);

## Coltivazione

INSERT INTO Coltivazione(id\_terreno,id\_prodottoterra)

VALUES('T1','C2'),

('T1','C4'),

('T1','C5'),

('T1','C6'),

('T1','C23'),

('T2','C7'),

('T2','C8'),

('T2','C9'),

('T2','C22'),

('T2','C26'),

('T2','C35'),

('T3','C28'),

('T3','C29'),

('T3','C30'),

('T3','C38'),

('T3','C39'),

('T3','C19'),

('T3','C20'),

('T4','C12'),

('T4','C13'),

('T4','C8'),

('T4','C9'),

('T4','C36'),

('T4','C37'),

('T5','C1'),

('T5','C11'),

('T5','C13'),

('T5','C30'),

('T6','C32'),

('T6','C33'),

('T7','C31'),

('T7','C40'),

('T7','C37'),

('T7','C33'),

('T7','C28'),

('T8','C12'),

('T8','C13'),

('T8','C29'),

('T8','C34'),

('T8','C18'),

('T8','C16'),

('T8','C10'),

('T9','C24'),

('T9','C25'),

('T10','C22'),

('T10','C34'),

('T10','C35'),

('T10','C36'),

('T10','C15'),

('T10','C16'),

('T11','C21'),

('T11','C5'),

('T11','C3'),

('T12','C19'),

('T12','C20'),

('T12','C33'),

('T12','C38'),

('T13','C6'),

('T13','C16'),

('T13','C17'),

('T13','C27'),

('T14','C27'),

('T14','C9'),

('T14','C13'),

('T15','C2'),

('T15','C24'),

('T15','C31'),

('T15','C39'),

('T15','C27'),

('T15','C4');

## Funzione

INSERT INTO Funzione(id\_funzione,nome\_funzione,descrizione)

VALUES ('F1','salute dei denti','aiuta la salute dei denti rinforzando lo smalto'),

('F2','salute delle ossa','aiuta la salute delle ossa rinforzandole'),

('F3','migliora metabolismo','aiuta l\'organismo a bruciare maggiori quantità di grassi'),

('F4','migliora metabolismo proteico','aiuta l\'organismo ad assimilare meglio le proteine'),

('F5','equilibrio idrosalino','grazie a questo la concentrazione di soluti viene mantenuta costante nel tempo'),

('F6','funzionamento muscoli','aiuta al rinforzamento dei tessuti muscolari'),

('F7','salute sistema nervoso','ha un effetto calmante, energizzante e aiuta la memoria'),

('F8','regolazione pressione arteriosa','regola la circolazione del sangue nelle arterie'),

('F9','produzione ATP','produce energia per le cellule, portando ad un aumento generale dell\'energia e ad una riduzione dell\'affaticamento'),

('F10','respirazione cellule','facilita la respirazione cellulare aiutando il Ciclo di Krebs '),

('F11','Mantiene equilibrio elettrolitico',NULL),

('F12','produzione succhi gastrici',NULL),

('F13','migliora sistema immunitario','aiuta il sistema immunitario nella produzione di globuli bianchi e globuli rossi'),

('F14','fertilità','favorisce la maturazione negli adolescenti e della fertilità negli adulti'),

('F15','azione antiossidante','sostanza in grado di contrastare,rallentare o neutralizzare la formazione dei radicali dell\'ossigeno che si formano in seguito a reazioni di ossidazione'),

('F16','prevenisce invecchiamento','prevenisce l\'invecchiamento delle cellule'),

('F17','protegge sviluppo dell\'embrione',NULL),

('F18','sintesi dell\'emoglobina','la sostanza aiuta nella sintesi dell\'emoglobina che si occupa del trasporto dell\'ossigeno dai polmoni ai tessuti e dell\'anidride carbonica dai tessuti ai polmoni');

## SMSvolge

INSERT INTO SMSvolge(simbolo\_elementochimico,id\_funzione)

VALUES ('P','F1'),

('P','F2'),

('Mg','F2'),

('P','F3'),

('Mg','F3'),

('Na','F5'),

('K','F6'),

('K','F7'),

('Mg','F7'),

('K','F8'),

('Mg','F9'),

('Cl','F11'),

('Cl','F12'),

('Fe','F18');

## VSvolge

INSERT INTO VSvolge(nome\_comunevitamina,id\_funzione)

VALUES ('A','F1'),

('A','F2'),

('D','F2'),

('B3','F3'),

('B5','F3'),

('B8','F4'),

('K','F8'),

('B3','F8'),

('B1','F9'),

('B2','F9'),

('B3','F10'),

('D','F13'),

('B6','F13'),

('A','F14'),

('E','F15'),

('B3','F15'),

('B6','F16'),

('B9','F17');

## Dieta

INSERT INTO Dieta(id\_dieta,nome\_dieta,durataraccomandata\_ingiorni,descrizione)

VALUES('D1','dieta rinforzante','14','dieta per rinforzare i tessuti e le ossa'),

('D2','dieta per la gravidanza','60','dieta adatta alle future mamme dalla terza settimana in poi di gravidanza'),

('D3','dieta per allenamento','21','dieta adatta per chi va in palestra o fa attività fisica giornalmente'),

('D4','dieta contro la spossatezza','a scelta','dieta che può aiutare, assieme ad una bilanciata alimentazione, ad avere più energia durante la giornata e diminuire i momenti di indebolimento'),

('D5','dieta riquilibrante','7','dieta che aiuta a ripristinare il normale metabolismo del corpo'),

('D6','dieta per le difese immunitarie','14','dieta che aiuta il corpo a costruire le proprie difese immunitarie, utile soprattutto nei periodi influenzali'),

('D7','dieta anti-invecchiamento','a scelta','dieta che aiuta a rallentare l\'invecchiamento del corpo'),

('D8','dieta anti-problemi circolatori','7','dieta che favorisce la circolazione sanguigna e aiuta a prevenire problemi cardio-circolatori');

## VUsataIn

INSERT INTO VUsataIn(id\_dieta,nome\_comunevitamina,quantitàgiornaliera\_ingrammi)

VALUES('D1','A','0.0005'),

('D1','D','0.00003'),

('D2','B9','0.00035'),

('D4','B1','0.01'),

('D4','B2','0.015'),

('D5','B3','0.04'),

('D6','D','0.000025'),

('D6','B6','0.01'),

('D7','B6','0.007'),

('D8','K','0.00002'),

('D8','B3','0.03');

## SMUsatoIn

INSERT INTO SMUsatoIn(id\_dieta,simbolo\_elementochimico,quantitàgiornaliera\_ingrammi)

VALUES('D1','P','1'),

('D1','Mg','0.3'),

('D3','Mg','0.35'),

('D3','K','1.2'),

('D4','Mg','0.3'),

('D5','P','0.8'),

('D5','Mg','0.2'),

('D8','Fe','0.0015'),

('D8','K','1.5');

## EffettuataDa

INSERT INTO EffettuataDa(CF,id\_dieta,inizio\_dieta,fine\_dieta)

VALUES('MNNGRM90P17H501D','D5','2022-11-05','2022-11-13'),

('MNNGRM90P17H501D','D1','2022-11-17','2022-11-23'),

('BRTMRM90P16H501D','D2','2022-09-19','2022-11-15'),

('HDZANT80P18H501D','D4','2022-07-19','2022-07-31'),

('HDZANT80P18H501D','D6','2022-12-12','2022-12-24'),

('HDZANT80P18H501D','D7','2022-07-19','2022-08-20'),

('HDZIVO68P18H501D','D1','2022-07-24','2022-12-10'),

('CMNRSL99P11H501D','D4','2022-09-11','2022-09-21'),

('CMNRSL99P11H501D','D8','2022-11-01','2022-11-08'),

('CSTDVD75P19H501D','D7','2022-12-01',NULL),

('XUUFRC84P22H501D','D3','2022-11-02','2022-11-28'),

('CMNGLA91P11H501D','D3','2022-09-21','2022-10-01'),

('CMNGLA91P11H501D','D3','2022-10-21','2022-11-01'),

('CMNGLA91P11H501D','D3','2022-11-21','2022-12-01'),

('NVCSRA99P17H501D','D5','2022-09-07','2022-09-15'),

('PRSRTA40P12H501D','D7','2022-10-18',NULL),

('PRSRTA40P12H501D','D8','2022-10-1','2022-10-8'),

('ATNFDR99P11H501D','D4','2022-07-12','2022-07-19'),

('MRCANN98P21H501D','D1','2022-08-14','2022-08-28'),

('MRCANN98P21H501D','D5','2022-10-07','2022-10-14'),

('ACCMRI79P11H501D','D4','2022-10-15','2022-10-31'),

('ACCMRI79P11H501D','D6','2022-11-07','2022-11-28');