

# **Progetto di Basi di Dati**

Riccardo Polini – Francesco Barbarulo

# Indice

1. Specifiche di Progetto .....	2
1.8. Indice di Manutenzione .....	4
1.8.1. Indice di Accrescimento Aereo & Radicale .....	4
1.8.3. Indice di Credibilità .....	6
1.8.4. Distanza Minima .....	6
1.9. Diagramma ER .....	9
2. Sviluppo del Progetto .....	10
2.1. Individuazione delle Ridondanze .....	14
2.2. Operazioni da compiere .....	14
2.3. Tavola dei volumi .....	17
2.4. Tavola degli accessi .....	20
2.5. Progettazione logica .....	28
2.6. Specifica dei vincoli d'integrità generici .....	30
2.7. Specifica dei vincoli d'integrità referenziale .....	32
2.8. Normalizzazione .....	35
3. Trigger ed Eventi .....	40
4. Area Analytics .....	47
4.1. Smart Design .....	47
4.2. Reporting .....	48
4.3. Indagini Statistiche .....	50

## **1. Specifiche di progetto**

La base di dati deve mantenere le informazioni necessarie al funzionamento del sistema informatico di una grande impresa distribuita sul territorio nazionale che si occupa di giardinaggio.

Nello specifico, la base di dati deve mantenere le seguenti informazioni:

### **1.1. Informazioni sulle Piante e sulle loro esigenze**

La base di dati mantiene informazioni sui tipi di piante vendute dall'azienda. Inoltre, per sapere di cosa ha bisogno un particolare tipo di pianta, vengono memorizzate anche loro esigenze in termini di luce, temperatura, terreno, irrigazione e concimazione. È richiesta, quindi, la memorizzazione delle seguenti informazioni:

- Codice pianta (codice identificativo per ogni tipo di pianta a cui appartengono più esemplari)
- Nome
- Cultivar (identifica una variante della stessa pianta)
- Sempreverde
- Dioica
- Indice accrescimento aereo
- Indice accrescimento radici
- Indice manutenzione

### **1.2. Informazioni sulle Patologie e sul loro trattamento**

In determinati periodi dell'anno, una pianta è suscettibile a particolari patologie, con una certa probabilità e entità. Ciascuna patologia è caratterizzata da sintomi che sono evidenti semplicemente osservando la pianta. Per questo, l'azienda si avvale di droni muniti di microcamere che riescono a individuare visivamente le principali patologie confrontando le immagini che raccolgono con quelle inserite precedentemente nella base di dati. I sintomi di ogni patologia sono classificati nella base di dati mediante:

- Codice
- Descrizione
- Immagine

Per ciascuna patologia, devono essere memorizzate le informazioni riguardo i trattamenti da somministrare:

- Codice prodotto
- Principio attivo
- Periodo vietato
- Tipo di somministrazione che può avvenire per irrigazione nel terreno o nebulizzazione o entrambi
- Tipologia (lotta chimica o lotta batterica)
- Tempo di attesa affinché il prodotto chimico usato si dissolva e sia divenuto innocuo per la salute dell'uomo

### **1.3. Informazioni sulle Serre**

L'azienda è dislocata sul territorio nazionale in più sedi, le quali si compongono generalmente di più serre. Per ciascuna serra, la base di dati deve memorizzare le seguenti informazioni:

- Codice Serra
- Nome
- Indirizzo
- Dimensioni (altezza, lunghezza, larghezza)
- Numero massimo di piante ospitali
- Numero di piante attualmente presenti

### **1.4. Informazioni sugli Utenti e sugli Ordini**

L'azienda ha un sito web che ospita una piattaforma social mediante la quale, gli utenti registrati possono interagire e acquistare esemplari di piante per il giardino. Ogni utente che si iscrive deve fornire alcune informazioni obbligatorie (nome, cognome, nickname, email, domanda segreta per il recupero della password con relativa risposta, città di residenza). Una volta in possesso di un profilo, il cliente può collegarsi allo store online ed effettuare ordini. Per ciascun ordine, la base di dati deve mantenere le seguenti informazioni:

- Codice dell'ordine
- Codice della pianta acquistata
- Codice del cliente
- Data di effettuazione dell'ordine
- Stato dell'ordine (in processazione, in preparazione, spedito, evaso)

### **1.5. Informazioni sulla Manutenzione**

L'azienda, oltre a svolgere la manutenzione delle piante in serra, offre al cliente interventi di manutenzione su richiesta, programmata e automatica. Gli interventi di manutenzione da effettuare (o già effettuati) sono memorizzati nel database e sono sempre visibile nel profilo del cliente.

### **1.6. Informazioni sul Forum online**

Il sito web dell'azienda contiene anche un'area social, un forum di discussione, nel quale gli utenti possono inserire post circa l'acquisto di nuove piante, per la cura di patologie che coinvolgono esemplari acquistati, o per condividere le loro esperienze. I post possono ricevere risposte da altri utenti del forum, le quali possono essere valutate con un giudizio.

### **1.7. Informazioni sulla progettazione di spazi verdi**

Oltre alla coltivazione delle piante, l'impresa si occupa anche di progettazione di spazi verdi. In particolare, in base alle caratteristiche fisiche del terreno, alla morfologia degli spazi e alle condizioni climatiche dell'area geografica fornite dal cliente mediante il sito web, il sistema informativo è in grado di fornire soluzioni che consistono di combinazioni di piante che potrebbero esservi messe a dimora, tenendo in considerazione sia l'accrescimento delle stesse che i futuri interventi di manutenzione.

### 1.8. Indice di Manutenzione

Abbiamo deciso di attribuire un valore numerico da 1 a 5 come indice di manutenzione secondo la seguente tabella:

Indice	Significato	Piante (alcune)
1	Nessuna manutenzione ordinaria è richiesta.	Abete, Larice, Quercia...
2	Manutenzione ordinaria ridotta al minimo, con un intervento di potatura annuale.	Platano, Pino...
3	Un intervento annuale e particolare attenzione alla concimazione e all'irrigazione.	Geraneo, Tulipano, Rosa...
4	Massimo due potature all'anno e particolare attenzione alle patologie.	Siepi, Cespugli...
5	Manutenzione ordinaria comprensiva di almeno due potature all'anno con particolare attenzione rivolta all'irrigazione, alle condizioni atmosferiche e alle patologie.	Limone, Arancio, Mandarino, Olivo, Pomodoro...

#### 1.8.1. Indice di Accrescimento Aereo

Indice	h = Altezza prevista (in metri)	Piante (alcune)
1	$h < 1$	Tutti i fiori, cespugli e ortaggi
2	$1 < h < 5$	Olivo, Limone, Arancio, Mandarino, Fico
3	$5 < h < 8$	Palma da datteri
4	$8 < h < 15$	Larice, Nespola, Palma da cocco
5	$h > 15$	Platano, Pino, Abete, Quercia, Faggio, Ciliegio

#### 1.8.2. Indice di Accrescimento Radicale

<b>Indice</b>	<b>Piante (alcune)</b>
1	Fiori, ortaggi e cespugli
2	Biancospino, Ligustro
3	Orniello, Limone, Arancio,
4	Cipresso, Acero, Betulla
5	Platano, Quercia, Cedro, Tiglio

### 1.8.3. Indice di Credibilità di un Utente

Presi in considerazione i seguenti dati:

NP : numero dei post pubblicati dall'utente,

NP<sub>tot</sub> : il numero dei post totali presenti nel database

NS : il numero di stelle ricevute sulle proprie risposte

NS<sub>tot</sub> : il massimo numero di stelle che un utente può ricevere in base ai post risposta pubblicati (5\*numRisposte)

L'indice di credibilità IC viene calcolato nel seguente modo:

$$IC = \frac{1}{2} \left[ \frac{NP \cdot 100}{NP_{tot}} + \frac{NS \cdot 100}{NS_{tot}} \right] [\%]$$

### 1.8.4. Distanza minima

Prima di analizzare la procedura che, data una pianta, restituisce la lista di piante con la relativa distanza minima alla quale posizionarle, di seguito sono elencate le view e le funzioni utilizzate per tale scopo.

La funzione *elementiNecessariTotali*, data una pianta, ne restituisce il numero di elementi richiesti.

```
DROP FUNCTION IF EXISTS elementiNecessariTotali;

DELIMITER $$
CREATE FUNCTION elementiNecessariTotali(_pianta CHAR(50))
RETURNS INT(11) DETERMINISTIC
BEGIN
    DECLARE numeroElementi INT(11) DEFAULT 0;

    SELECT COUNT(*) INTO numeroElementi
    FROM elementipiante
    WHERE codicePianta = _pianta;

    RETURN (numeroElementi);
END $$
DELIMITER ;
```

Fa utilizzo della view *elementiPiante*, la quale, per ogni pianta, seleziona una lista di elementi necessari.

```
CREATE OR REPLACE VIEW elementiPianta AS
SELECT codicePianta, infestante, sigla
FROM pianta NATURAL JOIN terreno NATURAL JOIN contiene1;
```

La procedura completa è la seguente:

```
DROP PROCEDURE IF EXISTS calcolaDistanzaMinima;

DELIMITER $$
CREATE PROCEDURE calcolaDistanzaMinima(_pianta CHAR(50))
BEGIN
    SET @piantaValida = (SELECT codicePianta
                        FROM Pianta
                        WHERE codicePianta = _pianta);

    IF @piantaValida IS NULL THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Codice pianta non valido';
    END IF;

    SET @oreLuce = (SELECT AVG(oreLuce)
                  FROM Pianta NATURAL JOIN Periodo NATURAL JOIN esigenze
                  WHERE codicePianta = @piantaValida);

    SET @indiceAccrAerea = (SELECT indiceAccrAerea
                           FROM Pianta
                           WHERE codicePianta = @piantaValida);

    SET @infestante = (SELECT infestante
                     FROM Pianta
                     WHERE codicePianta = @piantaValida);

    CREATE OR REPLACE VIEW elementiPianta AS
    SELECT codicePianta, infestante, sigla
    FROM Pianta NATURAL JOIN Terreno NATURAL JOIN contiene1;

    CREATE TEMPORARY TABLE IF NOT EXISTS distanzaParziale(
        codicePianta CHAR(50) NOT NULL,
        distanza INT(11) NOT NULL,
        PRIMARY KEY (codicePianta)
    );

    TRUNCATE TABLE distanzaParziale;

    INSERT INTO distanzaParziale
    SELECT P.codicePianta,
        CASE
            WHEN P.infestante = 1 OR @infestante = 1 THEN 10
            WHEN IFNULL(D.elementiInComune, 0)/elementiNecessariTotali(P.codicePianta)*100
BETWEEN 0 AND 30 THEN 1
            WHEN IFNULL(D.elementiInComune, 0)/elementiNecessariTotali(P.codicePianta)*100
BETWEEN 31 AND 60 THEN 3
            WHEN IFNULL(D.elementiInComune, 0)/elementiNecessariTotali(P.codicePianta)*100 >
60 THEN 5
        END AS distanzaMinima
    FROM Pianta P
        NATURAL JOIN Esigenze E
        LEFT OUTER JOIN (
```



```

SELECT EP1.codicePianta, COUNT(*) AS elementiInComune
FROM elementiPiante EP1
    INNER JOIN
        elementiPiante EP2 ON EP1.sigla = EP2.sigla
WHERE EP2.codicePianta = @piantaValida
GROUP BY EP1.codicePianta) AS D ON D.codicePianta = P.codicePianta
GROUP BY P.codicePianta;

SELECT DP.codicePianta,
CASE
    WHEN AVG(E.oreLuce) < 2 OR @oreLuce < 2 THEN DP.distanza
    WHEN (AVG(E.oreLuce) BETWEEN 2 AND 6 AND @indiceAccrAerea < 3) OR
        (@oreLuce BETWEEN 2 AND 6 AND P.indiceAccrAerea < 3) THEN IF(DP.distanza > 2,
DP.distanza, 2)
    WHEN (AVG(E.oreLuce) BETWEEN 2 AND 6 AND @indiceAccrAerea >= 3) OR
        (@oreLuce BETWEEN 2 AND 6 AND P.indiceAccrAerea >= 3) THEN IF(DP.distanza > 4,
DP.distanza, 4)
    WHEN (AVG(E.oreLuce) > 6 AND @indiceAccrAerea > 3) OR
        (@oreLuce > 6 AND P.indiceAccrAerea >= 3) THEN IF(DP.distanza > 5, DP.distanza, 5)
    WHEN (AVG(E.oreLuce) > 6 AND @indiceAccrAerea < 3) OR
        (@oreLuce > 6 AND P.indiceAccrAerea < 3) THEN IF(DP.distanza > 3, DP.distanza, 3)
END AS distanzaMinima
FROM distanzaParziale DP NATURAL JOIN Pianta P NATURAL JOIN Periodo PE NATURAL JOIN
Esigenze E
GROUP BY DP.codicePianta;
END $$
DELIMITER ;

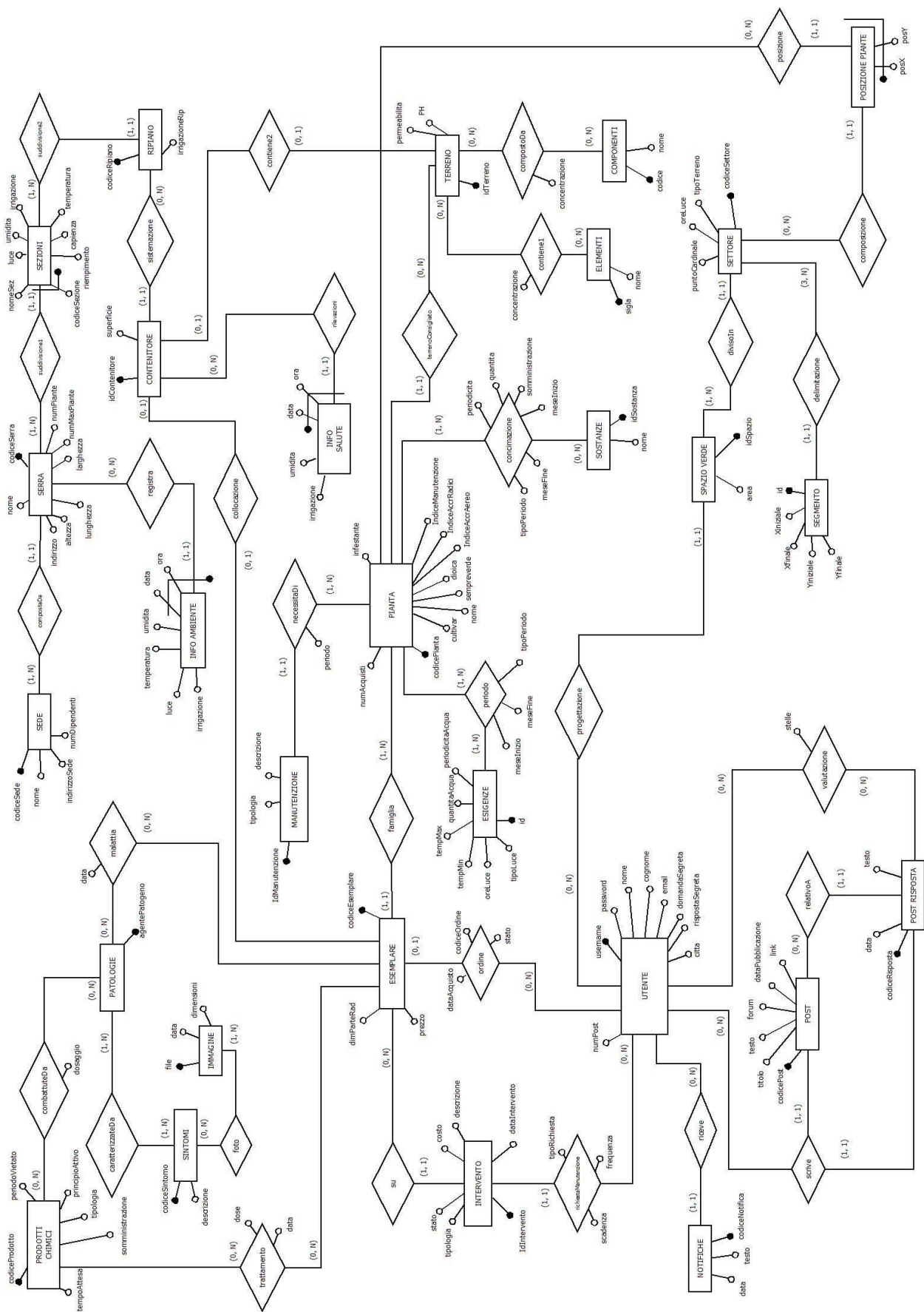
```

La procedura riceve in ingresso il codice di una pianta e controlla che sia valido. Se è valido ne ricava le ore di luce necessarie in media durante l'anno, l'indice di accrescimento della parte aerea e se è infestante o meno. Dopodiché viene creata una tabella temporanea dove sono inserite le piante e le distanze parziali in base agli elementi in comune alla pianta in ingresso. Con l'ultima selezione, la distanza parziale precedentemente calcolata può essere aggiornata in base alle ore di luce e all'indice di accrescimento della parte aerea delle piante prese in considerazione.

Un esempio di chiamata è il seguente:

```
CALL calcolaDistanzaMinima('000');
```

## 1.9. Diagramma E/R



## **2 Sviluppo del Progetto**

Sviluppiamo il progetto seguendo i punti indicati nelle specifiche.

Nel diagramma E/R si possono vedere:

- Entità e attributi con relativi identificatori primari
- Associazioni e cardinalità

Nello specifico, le associazioni sono:

### **Associazione Famiglia**

- Cardinalità (1, 1) con Esemplare, in quanto ciascun esemplare appartiene ad una sola famiglia.
- Cardinalità (1, N) con Pianta, in quanto ciascuna pianta fa riferimento ad almeno una famiglia.

### **Associazione necessitaDi**

- Cardinalità (1, 1) con Manutenzione, in quanto ciascuna manutenzione è relativa ad una sola specie.
- Cardinalità (1, N) con Pianta, in quanto ciascuna pianta può avere bisogno di uno o più interventi prestabiliti.

### **Associazione Ordine**

- Cardinalità (0, 1) con Esemplare, in quanto ciascun esemplare può essere acquistato da al più un utente.
- Cardinalità (0, N) con Utente, in quanto un utente può acquistare più esemplari (oppure nessun esemplare).

### **Associazione su**

- Cardinalità (0, N) con Esemplare, in quanto ciascun esemplare può ricevere più interventi di manutenzione (oppure nessun intervento).
- Cardinalità (1, 1) con Intervento, in quanto ciascun intervento di manutenzione fa riferimento ad uno e un solo esemplare.

### **Associazione RichiestaManutenzione**

- Cardinalità (0, N) con Utente, in quanto ciascun utente può richiedere più interventi di manutenzione (oppure nessuna manutenzione).
- Cardinalità (1, 1) con Intervento, in quanto ciascun intervento di manutenzione può essere richiesto da uno e un solo utente.

### **Associazione Scrive**

- Cardinalità (0, N) con Utente, in quanto ciascun utente può scrivere più post (oppure nessun post).
- Cardinalità (1, 1) con Post, in quanto ciascun post può essere scritto da uno e un solo utente.

### **Associazione Riceve**

- Cardinalità (0, N) con Utente, in quanto ciascun utente può ricevere più notifiche (oppure nessuna notifica).
- Cardinalità (1, 1) con Notifiche, in quanto ciascuna notifica può essere ricevuta da uno e un solo utente.

### **Associazione relativoA**

- Cardinalità (0, N) con Post, in quanto ciascun post può avere più post risposta oppure nessuno.
- Cardinalità (1, 1) con PostRisposta, in quanto ciascun postrisposta è relativo ad un solo post.

### **Associazione Valutazione**

- Cardinalità (0, N) con Utente, in quanto ciascun utente può valutare più postriposta (oppure nessun postriposta).
- Cardinalità (0, N) con PostRisposta, in quanto ciascun postriposta può essere valutato da più utenti (oppure nessun utente).

### **Associazione Progettazione**

- Cardinalità (0, N) con Utente, in quanto ciascun utente può progettare più spazi verdi (oppure nessuno spazio verde).
- Cardinalità (1, 1) con SpazioVerde, in quanto ciascuno spazio verde può essere progettato da uno e un solo utente.

### **Associazione DivisoIn**

- Cardinalità (1, N) con SpazioVerde, in quanto ciascun spazio verde è diviso in almeno un settore.
- Cardinalità (1, 1) con Settore, in quanto ciascun settore appartiene ad uno e un solo spazio verde.

### **Associazione Composizione**

- Cardinalità (0, N) con Settore, in quanto ciascun settore può contenere più piante (oppure nessuna pianta).
- Cardinalità (1, 1) con PosizionePianta, in quanto ciascuna pianta può essere inserita in uno e un solo settore.

### **Associazione Delimitazione**

- Cardinalità (3, N) con Settore, in quanto ciascun settore può essere delimitato da almeno 3 segmenti.
- Cardinalità (1, 1) con Segmento, in quanto ciascun segmento verde può appartenere ad uno e un solo settore.

### **Associazione Posizione**

- Cardinalità (1, 1) con PosizionePianta in quanto una posizione è relativa ad una ed una sola pianta.
- Cardinalità (0, N) con Pianta in quanto una pianta può avere o meno una posizione in uno spazio progettato.

### **Associazione Periodo**

- Cardinalità (1, N) con Pianta, in quanto ciascuna pianta ha almeno una esigenza in un certo periodo.
- Cardinalità (1, N) con Esigenze, in quanto ciascuna esigenza appartiene ad almeno una pianta.

### **Associazione Concimazione**

- Cardinalità (1, N) con Pianta, in quanto ciascuna pianta ha bisogno di essere concimata con almeno una sostanza.
- Cardinalità (0, N) con Sostanze, in quanto ciascuna sostanza può essere presente nella concimazione di più piante (oppure nessuna pianta).

### **Associazione TerrenoCosigliato**

- Cardinalità (1, 1) con Pianta, in quanto ciascuna pianta può avere uno e un solo terreno consigliato.
- Cardinalità (0, N) con Terreno, in quanto ciascun tipo di terreno può essere consigliato per più piante (oppure nessuna pianta).

**Associazione Contiene1**

- Cardinalità (0, N) con Terreno, in quanto ciascun tipo di terreno può contenere più elementi (oppure nessuno elemento).
- Cardinalità (0, N) con Elementi, in quanto ciascun elemento può caratterizzare più tipi di terreno (oppure nessun tipo di terreno).

**Associazione CompostoDa**

- Cardinalità (1, N) con Terreno, in quanto ciascun tipo di terreno è composto da almeno un componente.
- Cardinalità (0, N) con Componenti, in quanto ciascun componente può caratterizzare più tipi di terreno (oppure nessun tipo di terreno).

**Associazione Contiene2**

- Cardinalità (1, N) con Terreno, in quanto ciascun tipo di terreno può essere contenuto in al più un contenitore.
- Cardinalità (0, 1) con Contenitore, in quanto ciascun contenitore può essere riempito con al massimo un tipo di terreno.

**Associazione Collocazione**

- Cardinalità (0, 1) con Esempio, in quanto ciascun esemplare è collocato in un solo contenitore, oppure in nessuno qualora sia stato venduto.
- Cardinalità (0, 1) con Contenitore, in quanto ciascun contenitore può contenere al massimo un esemplare.

**Associazione Sistemazione**

- Cardinalità (1, 1) con Contenitore, in quanto ciascun contenitore è sistemato su uno e un solo ripiano.
- Cardinalità (0, N) con Ripiano, in quanto ciascun ripiano può accogliere più contenitori (oppure nessun contenitore).

**Associazione Suddivisione2**

- Cardinalità (1, 1) con Ripiano, in quanto ciascun ripiano appartiene ad una ed una sola sezione.
- Cardinalità (1, N) con Sezioni, in quanto ciascuna sezione può essere suddivisa in almeno un ripiano.

**Associazione Suddivisione1**

- Cardinalità (1, N) con Serra, in quanto ciascuna serra può essere suddivisa in almeno una sezione.
- Cardinalità (0, 1) con Sezioni, in quanto ciascuna sezione appartiene ad una e una sola serra.

**Associazione Registra**

- Cardinalità (0, N) con Serra, in quanto in ciascuna serra possono essere registrate più informazioni sull'ambiente (oppure nessuna informazione).
- Cardinalità (1, 1) con InfoAmbiente, in quanto ciascuna informazione sull'ambiente è stata registrata in una determinata serra.

**Associazione CompostaDa**

- Cardinalità (1, N) con Sede, in quanto ciascuna sede ha almeno una serra.

- Cardinalità (1, 1) con Serra, in quanto ciascuna serra appartiene ad una ed una sola sede.

#### **Associazione Malattia**

- Cardinalità (0, N) con Esemplare, in quanto ciascun esemplare può essere affetta da più patologie (oppure da nessuna patologia).
- Cardinalità (0, N) con Patologie, in quanto ciascuna patologia può interessare più esemplari (oppure nessuno).

#### **Associazione CombattuteDa**

- Cardinalità (0, N) con ProdottiChimici, in quanto ciascun prodotto chimico può servire a combattere più patologie (oppure nessuna patologia).
- Cardinalità (0, N) con Patologie, in quanto ciascun Patologia può essere combattuta da più prodotti chimici (oppure nessun prodotto chimico).

#### **Associazione CaratterizzateDa**

- Cardinalità (1, N) con Patologie, in quanto ciascuna patologia è caratterizzata da almeno un sintomo.
- Cardinalità (1, N) con Sintomi, in quanto ciascun sintomo caratterizza almeno una patologia.

#### **Associazione Foto**

- Cardinalità (0, N) con Sintomi, in quanto ciascun sintomo può avere più foto (oppure nessuna foto).
- Cardinalità (1, N) con Immagine, in quanto ciascuna immagine appartiene ad almeno un sintomo.

#### **Associazione Trattamento**

- Cardinalità (0, N) con Esemplare in quanto ciascun esemplare può essere trattato con più prodotti chimici (oppure con nessuno).
- Cardinalità (0, N) con ProdottiChimici in quanto un prodotto può o non può essere utilizzato in uno o più trattamenti.

## 2.1. Individuazione delle ridondanze

Nello schema ER sono presenti le seguenti ridondanze:

### Ridondanza 1:

L'attributo *numAcquisti* di Pianta che contiene il numero di esemplari acquistati di ogni specie.

Viene aggiornato mediante un trigger che incrementa il conteggio ogni volta che viene acquistato un esemplare oppure ogni volta che un ordine "pendente" viene concluso.

### Ridondanza 2:

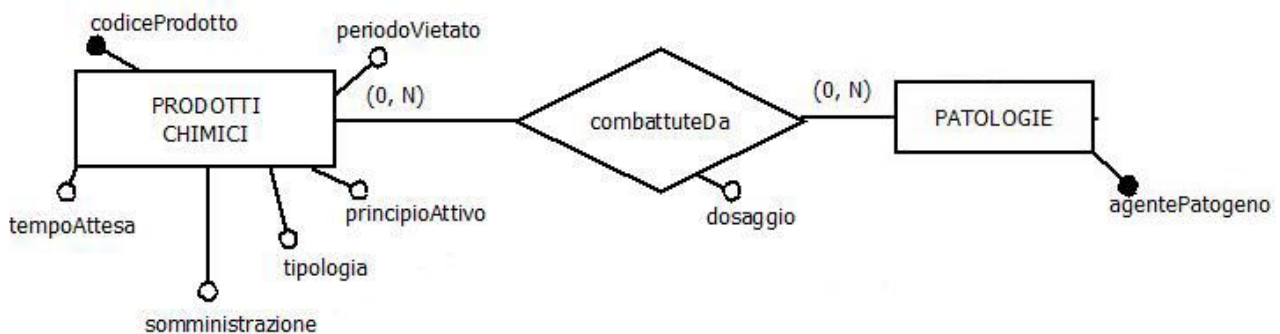
L'attributo *numPost* di Utente che contiene il numero totale di tutti i post pubblicati da un utente.

Viene aggiornato mediante un trigger che incrementa il conteggio ogni qualvolta un utente pubblica un post, oppure un post risposta.

## 2.2. Operazioni da compiere sulla Base di Dati

### Operazione 1

-Descrizione: Trovare il Prodotto Chimico adatto per combattere una determinata Patologia e tutte le altre informazioni su di esso.



-Frequenza: 50 volte al giorno.

-SQL:

```
SELECT PC.*
FROM prodottichimici P INNER JOIN combattuteda C
ON P.codiceProdotto=C.codiceProdotto
WHERE C.agentePatogeno='nome';
```

### Operazione 2

-Descrizione: Inserimento di nuovi esemplari nel database.

-Frequenza: 1 volta al giorno.

-SQL:

```
INSERT INTO Esemplari VALUES("valori passati");
```

### Operazione 3

-Descrizione: Visualizzare le Informazioni Ambientali di una data serra relative alla giornata odierna.

-Frequenza: 100 volte al giorno.

-SQL:

```
SELECT *  
FROM infoambiente  
WHERE codiceSerra='xxxx'  
AND data=CURRENT_DATE;
```

### Operazione 4

-Descrizione: Costruire lo storico degli ordini di un certo utente.

-Frequenza: 1 volta al giorno.

-SQL:

```
SELECT codiceOrdine, dataAcquisto, codiceEsemplare  
FROM ordine  
WHERE username='xxx'  
AND stato='evaso'  
ORDER BY dataAcquisto DESC;
```

### Operazione 5

-Descrizione: Registrare una nuova Info Salute relativa ad un contenitore.

-Frequenza: 3 volte al giorno

-SQL:

```
INSERT INTO infosalute VALUES('valori passati');
```

### Operazione 6

-Descrizione: Visualizzare quanti esemplari di una data specie sono stati venduti.

-Frequenza: 1 volta al giorno

-SQL:

```
SELECT nome, numAcquisti  
FROM pianta  
WHERE codicePianta='xxx';
```

### Operazione 7

-Descrizione: Annullare un intervento di Manutenzione

-Frequenza: 10 volte al giorno

-SQL:

```
DELETE FROM intervento  
WHERE idIntervento='xxxx';
```



### Operazione 8

- Descrizione: Visualizzare il numero totale di post pubblicati da un certo utente
- Frequenza: 5 volte al giorno
- SQL:

```
SELECT numPost  
FROM utente  
WHERE username='xxx';
```

### 2.3. Tavola dei volumi

Nome	E/R	Numero istanze	Motivazione
Pianta	E	250	Ipotesi iniziale
Esigenze	E	500	Ipotesi iniziale
terrenoConsigliato	R	250	Cardinalità (1,1) con Pianta
Concimazione	R	500	In media una pianta richiede due concimazioni differenti all'anno: $2*250=500$
Sostanze	E	30	Ipotesi iniziale
necessitaDi	R	500	Cardinalità (1,1) con Manutenzione
Manutenzione	E	500	In media una pianta ha 2 manutenzioni prestabilite all'anno: $250*2=500$
Periodo	R	1000	Una pianta in media ha due diverse esigenze di trattamento all'anno: $2*500=1000$
Terreno	E	100	Ipotesi iniziale
Contiene1	R	1000	In media un Terreno contiene una decina di elementi diversi: $10*100=1000$
Elementi	E	100	Ipotesi iniziale
compostoDa	R	90	Ci sono circa 3 componenti per ogni tipologia di Terreno: $30*3=90$
Componenti	E	30	Ipotesi iniziale
Malattia	R	2500000	Una esemplare può soffrire di al più due patologie contemporaneamente: $2*1250000=2500000$
Patologie	E	100	Ipotesi iniziale
Caratterizzazione	R	60	Una patologia in media è caratterizzata da due sintomi distinti: $2*30=60$
Sintomi	E	30	Ipotesi iniziale
Fotografia	R	200	Ipotesi iniziale
Immagine	E	100	Ipotesi iniziale
combattuteDa	R	300	Ipotesi iniziale
Prodotti Chimici	E	100	Ipotesi iniziale
Trattamento	R	125000	In media 1 esemplare su 10 necessita di un trattamento: $1250000/10 = 125000$ .

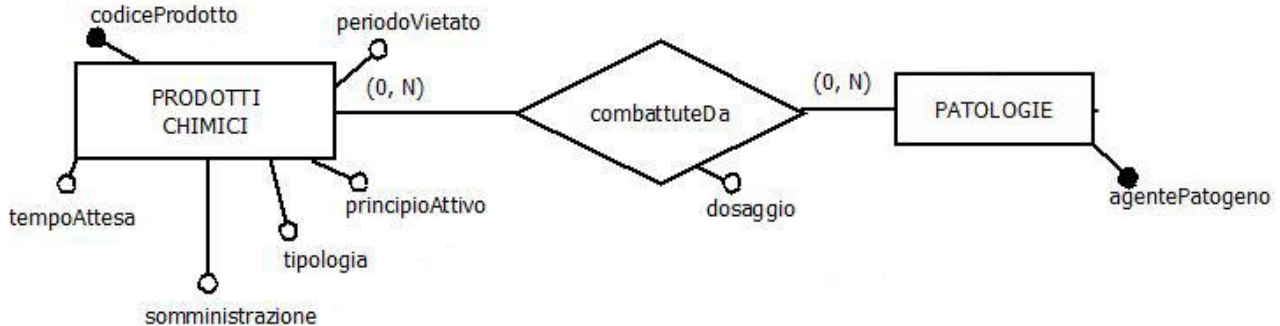
Collocazione	R	1210000	Poiché ci sono 40000 ordini, gli esemplari collocati in contenitori saranno: $1250000 - 40000 = 1210000$
Contenitore	E	1300000	Ipotesi iniziale
Rilevazioni	R	3750000	Cardinalità (1,1) con Info Salute
Info Salute	E	3750000	In media vengono fatte 3 rilevazioni al giorno per ogni contenitore con pianta: $3 * 1250000 = 3750000$
Contiene2	R	1250000	Si suppone che ci siano sempre 1250000 contenitori pieni
Sistemazione	R	1300000	Cardinalità (1,1) con Contenitore
Ripiano	E	2500	Si ipotizzano 5 ripiani per ogni sezione: $5 * 500 = 2500$
Suddivisione1	R	2500	Cardinalità (1,1) con Ripiano
Sezioni	E	500	In ogni serra ci sono 5 sezioni: $5 * 100 = 500$
Suddivisione2	R	500	Cardinalità (1,1) con Sezioni
Serra	E	100	Ci sono circa 5 serre per ogni sede: $5 * 20 = 100$
compostaDa	R	100	Cardinalità (1,1) con Serra
Sede	E	20	Una sede per ciascuna regione italiana
Registra	R	300	Cardinalità (1,1) con Info Ambiente
Info Ambiente	E	300	Si ipotizza che vengano registrate le informazioni ambientali 3 volte al giorno per ogni serra: $3 * 100 = 300$
Esemplare	E	1250000	Ci sono 50 esemplari per ogni tipo di pianta, per ogni serra $1250000 = (50 * 250) * 100$
ordine	R	40000	Ogni utente possiede in media 5 esemplari $40000 = 8000 * 5$
Utente	E	8000	Ipotesi iniziale
su	R	3750000	Cardinalità (1, 1) con Intervento
Intervento	E	3750000	Un esemplare può ricevere, in media, 3 interventi di manutenzione all'anno $3750000 = 1250000 * 3$
Famiglia	R	1250000	Cardinalità (1, 1) con Esemplare

richiestaManutenzione	R	3750000	Cardinalità (1, 1) con Intervento
Scrive	R	64000	Ipotesi iniziale
Post	E	64000	Cardinalità (1, 1) con Scrive
relativoA	R	192000	Cardinalità (1, 1) con Post Risposta
PostRisposta	E	192000	In media 1 post ha 3 postRisposta $192000 = 64000 * 3$
Valutazione	R	960000	Un utente, in media, valuta 5 postRiposta $960000 = 192000 * 5$
Progettazione	R	800	Uno spazio verde, in media, è progettato da 0.1 utenti $800 = 8000 * 0.1$
SpazioVerde	E	800	Cardinalità (1, 1) con Progettazione
divisoIn	R	4000	Uno spazio verde, in media, è suddiviso in 5 settori: $4000 = 800 * 5$
Settore	E	4000	Cardinalità (1, 1) con Suddivisione
Delimitazione	R	12000	Un settore, in media, è delimitato da 3 segmenti $12000 = 4000 * 3$
Segmento	E	12000	Cardinalità (1, 1) con Delimitazione
Composizione	R	12000	Un settore, in media, è composto da 3 esemplari $12000 = 4000 * 3$
PosizionePiante	E	12000	Cardinalità (1, 1) con Composizione
Posizione	R	12000	Cardinalità (1, 1) con PosizionePiante

## 2.4. Tavola degli accessi

### Operazione 1:

Trovare i Prodotti Chimici adatti per combattere una determinata Patologia e tutte le altre informazioni su di essi.



Nome	E/R	Istanze	Motivazione
combattuteDa	R	300	Per ogni agente patogeno ci sono, in media, 3 prodotti chimici adatti a combatterlo: $3 \times 100 = 300$
Prodotti Chimici	E	100	Ipotesi iniziale

Numero operazioni elementari	Tipo operazione	Costrutto	Nome Costrutto	Motivazione
$3 \times 1 = 3$	Lettura	R	combattuteDa	Leggo i 3 prodotti chimici adatti a combattere la patologia richiesta.
$3 \times 1 = 3$	Lettura	E	ProdottiChimici	Leggo le rispettive informazioni, per ogni prodotto chimico trovato.
$3 + 3 = 6$	Numero operazioni elementari per singola operazione			
$6 \times 50 = 300$	Totale operazioni elementari al giorno			

### Operazione 2:

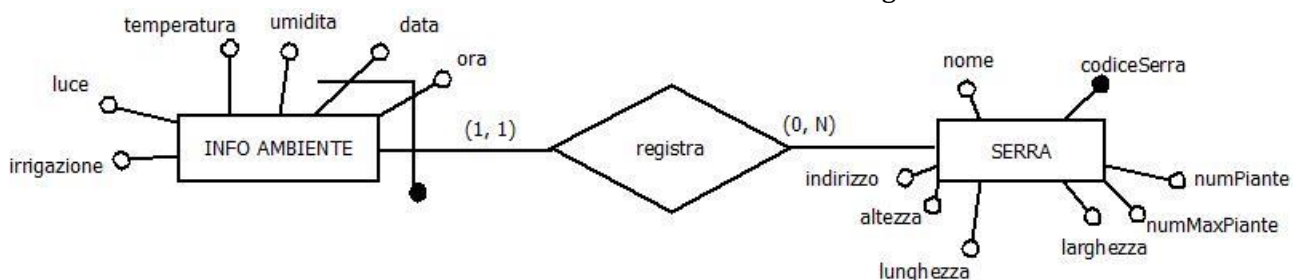
Inserimento di nuovi esemplari nel database.

Nome	E/R	Numero istanze	Motivazione
Sede	E	20	Una sede per ciascuna regione italiana

Numero operazioni elementari	Tipo operazione	Costrutto	Nome Costrutto	Motivazione
20x2=40	Scrittura	E	Esemplare	Si suppone che, ogni giorno, arriva una nuova partita di esemplari in ogni sede. Ogni sede, una volta ricevuta la nuova partita di esemplari, provvederà ad inserirli nel database. Poiché ci sono 20 sedi: 2x20=40
40	Numero operazioni elementari per singola operazione			
40x1=40	Totale operazioni elementari al giorno			

### Operazione 3:

Visualizzare le Informazioni Ambientali di una data serra relative alla giornata odierna.

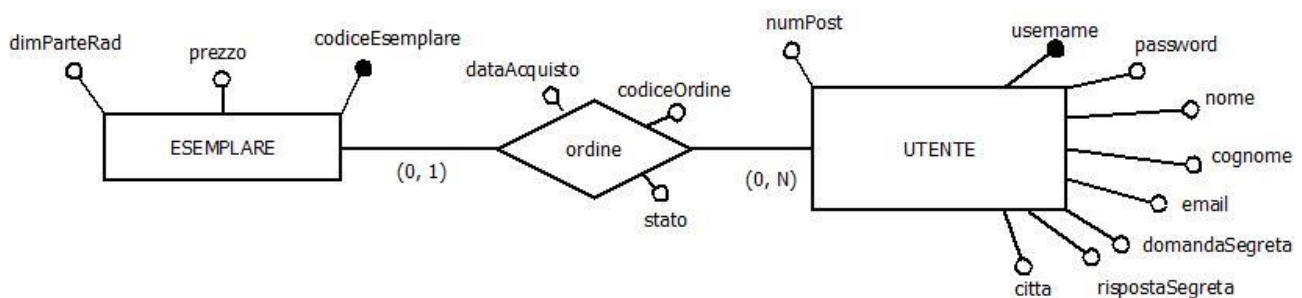


Nome	E/R	Numero istanze	Motivazione
Serra	E	100	Ci sono circa 5 serre per ogni sede: 5*20=100
Registra	R	300	Cardinalità (1,1) con Info Ambiente
Info Ambiente	E	300	Si ipotizza che vengano registrate le informazioni ambientali 3 volte al giorno per ogni serra: 3*100=300

Numero operazioni elementari	Tipo operazione	Costrutto	Nome Costrutto	Motivazione
3x1=3	Lettura	E	InfoAmbiente	Vengono registrate le informazioni ambientali 3 volte al giorno per ogni serra
3	Numero operazioni elementari per singola operazione			
3x100=300	Totale operazioni elementari al giorno			

#### Operazione 4:

Costruire lo storico degli ordini di un certo utente.

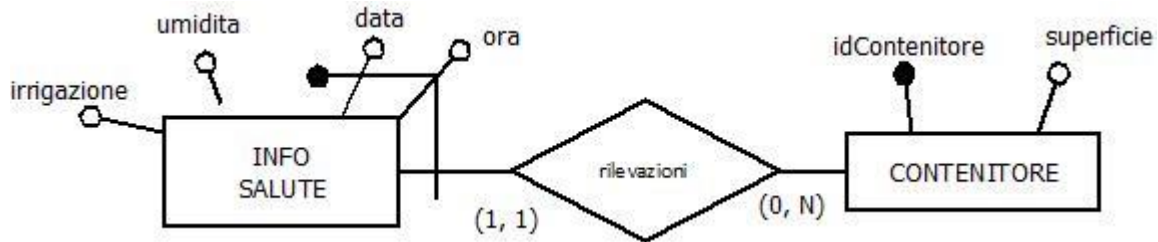


Nome	E/R	Numero istanze	Motivazione
ordine	R	40000	Ogni utente possiede in media 5 esemplari $40000 = 8000 * 5$
Utente	E	8000	Ipotesi iniziale
Esemplare	E	1250000	Ci sono 50 esemplari per ogni tipo di pianta, per ogni serra $1250000 = (50 * 250) * 100$

Numero operazioni elementari	Tipo operazione	Costrutto	Nome Costrutto	Motivazione
5x1=5	Lettura	R	Ordine	Un utente, in media, ha ordinato 5 esemplari. Devo quindi fare 5 letture su Ordine per recuperare tutti gli ordini effettuati dall'utente specificato.
5	Numero operazioni elementari per singola operazione			
5x1=5	Totale operazioni elementari al giorno			

#### Operazione 5:

Registrare una nuova Info Salute relativa ad un contenitore.

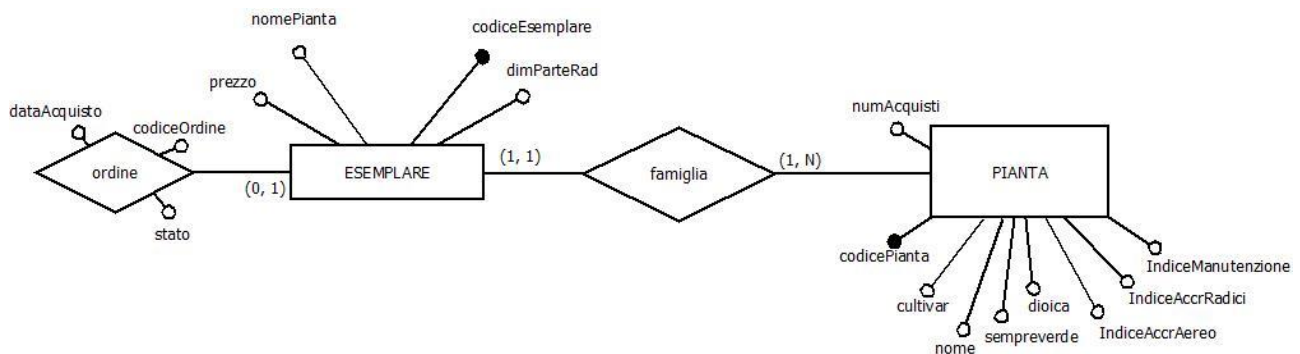


Nome	E/R	Numero istanze	Motivazione
Contenitore	E	1300000	Ipotesi iniziale
Rilevazioni	R	3750000	Cardinalità (1,1) con Info Salute
Info Salute	E	3750000	In media vengono fatte 3 rilevazioni al giorno per ogni contenitore con pianta: $3 \times 1250000 = 3750000$

Numero operazioni elementari	Tipo operazione	Costrutto	Nome Costrutto	Motivazione
1x2=2	Scrittura	E	Info Salute	Per ogni contenitore con pianta, vengono fatte 3 rilevazioni al giorno, ciascuna inserita immediatamente nel database.
2	Numero operazioni elementari per singola operazione			
2x3=6	Totale operazioni elementari al giorno			

#### Operazione 6:

Visualizzare quanti esemplari di una data specie sono stati venduti.





Nome	E/R	Numero istanze	Motivazione
Pianta	E	250	Ipotesi iniziale
Famiglia	R	1250000	Cardinalità (1, 1) con Esemplare
Esemplare	E	1250000	Ci sono 50 esemplari per ogni tipo di pianta, per ogni serra $1250000 = (50 \cdot 250) \cdot 100$
Ordine	R	40000	Ogni utente possiede in media 5 esemplari $40000 = 8000 \cdot 5$

Senza ridondanza 6				
Numero operazioni elementari	Tipo operazione	Costrutto	Nome Costrutto	Motivazione
$1 \times 35000 = 35000$	Lettura	R	Ordine	Si ipotizza che su 40000 ordini, 35000 siano conclusi.
$1 \times 140 = 140$	Lettura	E	Esemplare	Poiché ci sono 250 specie diverse di piante, per sapere quanti esemplari sono stati venduti di una singola specie bisogna fare $35000 / 250 = 140$
$35000 + 140 = 35140$	Numero operazioni elementari per singola operazione			
$35140 \times 1 = 35140$	Totale operazioni elementari al giorno			

Con ridondanza 6		
Numero operazioni elementari	Tipo operazione	Motivazione
$1 \times 1 = 1$	Lettura	Compio 1 operazione di lettura su Pianta per leggere l'attributo numAcquisti
$12 \times 2 = 24$	Scrittura	Costo giornaliero per mantenere aggiornata la ridondanza. Si suppone che vengano conclusi 12 ordini al giorno.
$24 + 1 = 25$	Totale operazioni elementari per singola operazione	
$25 \times 1 = 25$	Totale operazioni elementari al giorno	

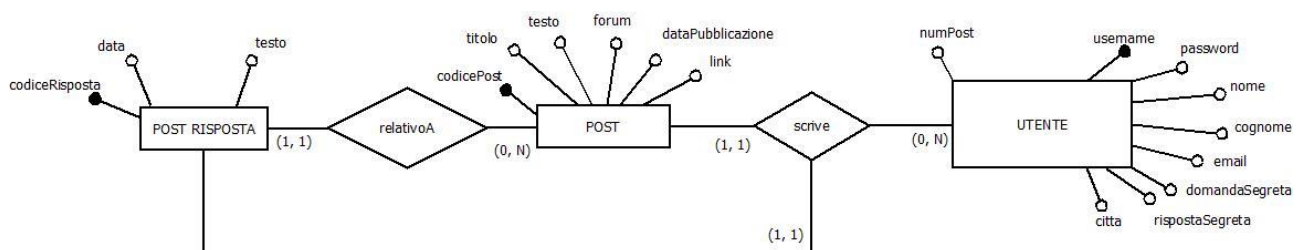
### Operazione 7:

Annullare un intervento di Manutenzione

Numero operazioni elementari	Tipo operazione	Costrutto	Nome Costrutto	Motivazione
$2 \times 1 = 2$	Scrittura	E	Intervento	Dato il codice dell'intervento che voglio annullare, cancello la rispettiva tupla.
2	Numero operazioni elementari per singola operazione			
$2 \times 10 = 20$	Totale operazioni elementari al giorno			

### Operazione 8:

Visualizzare il numero totale di post pubblicati da un certo utente



Nome	E/R	Numero istanze	Motivazione
Utente	E	8000	Ipotesi iniziale
Scrive	R	64000	Ipotesi iniziale
Post	E	64000	Cardinalità (1, 1) con Scrive
PostRisposta	E	192000	In media 1 post ha 3 postRisposta $192000 = 64000 \times 3$

Senza ridondanza 8				
Numero operazioni elementari	Tipo operazione	Costrutto	Nome Costrutto	Motivazione
8	Lettura	E	Post	Poiché ci sono 64000 post e 8000 utenti, un utente, in media, è attribuibile a $64000/8000 = 8$ post
24	Lettura	E	PostRisposta	Poiché ci sono 192000 post risposta e 8000 utenti, un utente, in media, è attribuibile a $192000/8000 = 24$ post risposta
32	Numero operazioni elementari per singola operazione			
$32 \times 5 = 160$	Totale operazioni elementari al giorno			

Con ridondanza 8		
Numero operazioni elementari	Tipo operazione	Motivazione
1	Lettura	Compio 1 operazione di lettura su Utente per leggere l'attributo numPost
$2 \times 1 = 2$	Scrittura	Costo giornaliero per mantenere aggiornata la ridondanza. Si suppone che un Utente pubblichi in media un post al giorno.
3	Totale operazioni elementari per singola operazione	
$3 \times 5 = 15$	Totale operazioni elementari al giorno	

La tavola degli accessi delle restanti operazioni non cambia in base alla presenza o meno delle ridondanze individuate.

Ora, in base ai risultati ottenuti, possiamo decidere se lasciare o meno le ridondanze analizzate.

Ridondanza 1:

Senza ridondanza	
Operazione	Numero operazioni elementari al giorno
Operazione 6	35140
<b>Totale:</b>	35140

Con ridondanza	
Operazione	Numero operazioni elementari al giorno
Operazione 6	25
<b>Totale:</b>	25

La ridondanza 1 riduce i tempi di esecuzione di quell'operazione. Si decide quindi di lasciare tale ridondanza.

Ridondanza 2:

Senza ridondanza	
Operazione	Numero operazioni elementari al giorno
Operazione 8	160
<b>Totale:</b>	160

Con ridondanza	
Operazione	Numero operazioni elementari al giorno
Operazione 8	15
<b>Totale:</b>	15

La ridondanza 2 riduce i tempi di esecuzione di quell'operazione. Si decide quindi di lasciare tale ridondanza.

## 2.5. Progettazione logica

- PIANTA (codicePianta, idTerreno, cultivar, nome, sempreverde, dioica, indiceAccrAereo, indiceAccrRadici, indiceManutenzione, infestante)
- PERIODO (id, codicePianta, tipoPeriodo, meseInizio, meseFine)
- ESIGENZE (id, tipoLuce, oreLuce, tempMin, tempMax, quantitaAcqua, periodicitAcqua)
- CONCIMAZIONE (idSostanza, codicePianta, tipoPeriodo, periodicit, quantita, somministrazione, meseInizio, meseFine)
- SOSTANZE (idSostanza, nome)
- TERRENO (idTerreno, permeabilita, PH)
- CONTIENE1 (idTerreno, sigla, concentrazione)
- CONTIENE2 (idContenitore, idTerreno)
- COMPOSTODA (idTerreno, codice, concentrazione)
- ELEMENTI (sigla, nome)
- COMPONENTI (codice, nome)
- ESEMPLARE (codiceEsemplare, idContenitore, codicePianta, prezzo, dimParteRad)
- ORDINE (username, codiceEsemplare, dataAcquisto, codiceOrdine, stato)
- UTENTE (username, password, nome, cognome, email, domandaSegreta, rispostaSegreta, citta)
- NOTIFICA (codiceNotifica, username, testo, data)
- POST (codicePost, username, titolo, testo, forum, dataPubblicazione, link)
- POSTRISPOSTA (codiceRisposta, username, codicePost, testo, dataPubblicazione)
- VALUTAZIONE (codiceRisposta, username, stelle)
- INTERVENTO (idIntervento, username, codiceEsemplare, tipoRichiesta, scadenza, costo, frequenza, tipologia, descrizione, stato, dataIntervento)
- MANUTENZIONE (idManutenzione, codicePianta, tipologia, periodo, descrizione)
- SPAZIOVERDE (id, username, area)
- SETTORE (codiceSettore, idSpazio, tipoTerreno, oreLuce, puntoCardinale)
- SEGMENTO (codiceSettore, idSpazio, Xiniziale, Yiniziale, Xfinale, Yfinale)
- POSIZIONE PIANTE (codiceSettore, posX, posY, codicePianta)
- MALATTIA (agentePatogeno, codiceEsemplare)
- PATOLOGIE (agentePatogeno)
- CARATTERIZZATEDA (codiceSintomo, agentePatogeno)
- SINTOMI (codiceSintomo, descrizione)
- FOTO (codiceSintomo, file)
- IMMAGINE (file, data, dimensioni)
- COMBATTUTEDA (agentePatogeno, codiceProdotto, dosaggio)
- PRODOTTI CHIMICI (codiceProdotto, periodoVietato, tempoAttesa, principioAttivo, tipologia, somministrazione)
- TRATTAMENTO (codiceProdotto, codiceEsemplare, codiceTrattamento, dose, data)
- CONTENITORE (idContenitore, codiceRipiano, superficie)
- INFOSALUTE (idContenitore, data, ora, umidita, irrigazione)
- RIPIANO (codiceRipiano, codiceSezione, codiceSerra, irrigazioneRip)
- SEZIONI (codiceSezione, codiceSerra, nomeSezione, riempimento, capienza, temperatura, luce, umidita, irrigazione)
- SERRA (codiceSerra, codiceSede, nome, indirizzo, altezza, larghezza, lunghezza, numMaxPiante, numMinPiante)

- INFOAMBIENTE (codiceSerra, data, ora, umidita, temperatura, luce, irrigazione)
- SEDE (codiceSede, nome, indirizzoSede, numDipendenti)

## 2.6. Specifica dei vincoli di integrità generici

Di seguito sono elencati i vincoli di integrità generici presenti e la rispettiva implementazione mediante trigger.

- L'attributo tipologia della tabella INTERVENTO può assumere soltanto valori tra quelli elencati di seguito: potatura, aumento produzione, rimozione, messa a terra, rinvaso, concimazione.
- L'attributo tipoRichiesta della tabella INTERVENTO può assumere soltanto valori tra quelli elencati di seguito: su richiesta, automatica, programmata.

### Trigger di controllo:

```
DROP TRIGGER IF EXISTS checkTipologia;

DELIMITER $$
CREATE TRIGGER checkTipologia
BEFORE INSERT ON Intervento FOR EACH ROW
BEGIN
    IF (NEW.tipoRichiesta <> 'su richiesta' AND NEW.tipoRichiesta <> 'automatica' AND
        NEW.tipoRichiesta <> 'programmata' AND NEW.tipoRichiesta <> 'autonoma') OR
        ((NEW.tipologia <> 'potatura') AND (NEW.tipologia <> 'rinvaso') AND (NEW.tipologia <>
        'concimazione') AND (NEW.tipologia <> 'messa a terra') AND (NEW.tipologia <> 'aumento
        produzione') AND (NEW.tipologia <> 'rimozione')) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Tipologia o tipo di richiesta non riconosciuta';
    END IF;
END$$
DELIMITER ;
```

- L'attributo stato della tabella ORDINE può assumere soltanto valori tra quelli elencati di seguito, nella stessa sequenza: pendente, in processazione, in preparazione, spedito, evaso.

### Trigger di controllo:

```
DROP TRIGGER IF EXISTS checkStato;

DELIMITER $$
CREATE TRIGGER checkStato
BEFORE UPDATE ON Ordine FOR EACH ROW
BEGIN
    IF (NEW.Stato <> 'pendente' AND NEW.Stato <> 'processazione' AND NEW.Stato <> 'preparazione'
        AND NEW.Stato <> 'spedito' AND NEW.Stato <> 'evaso') THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Stato inserito non errato';
    ELSE
        UPDATE Ordine
        SET stato = NEW.Stato
        WHERE CodiceOrdine = NEW.CodiceOrdine;
```

```
END IF;  
END$$  
DELIMITER ;
```

*Esiste un altro trigger identico a questo sull'INSERT nella medesima tabella, il quale, però, controlla che lo stato inserito sia uno tra 'pendente' e 'processazione', poiché possono essere gli stati di partenza di un ordine.*

**Buisness rule che impedisce di inserire un valore non permesso nell'attributo *quantitaAcqua* della tabella **ESIGENZE**:**

```
DROP TRIGGER IF EXISTS controlloQuantitaAcqua;  
  
DELIMITER $$  
  
CREATE TRIGGER controlloQuantitaAcqua  
BEFORE INSERT ON esigenze FOR EACH ROW  
BEGIN  
    IF((NEW.quantitaAcqua <> 'basso') AND (NEW.quantitaAcqua <> 'medio') AND  
(NEW.quantitaAcqua <> 'alto')) THEN  
        SIGNAL SQLSTATE '45000'  
        SET MESSAGE_TEXT= 'ERRORE: Valore quantità acqua non consentito.';  
    END IF;  
END$$  
DELIMITER ;
```



## 2.7. Specifica dei vincoli di integrità referenziale

Di seguito sono elencati i vincoli di integrità referenziale presenti nella base di dati creata.

- Esiste un vincolo di integrità referenziale tra l'attributo CodiceSintomo della tabella CARATTERIZZATEDA e l'attributo CodiceSintomo della tabella SINTOMI.
- Esiste un vincolo di integrità referenziale tra l'attributo AgentePatogeno della tabella CARATTERIZZATEDA e l'attributo AgentePatogeno della tabella PATOLOGIE
- Esiste un vincolo di integrità referenziale tra l'attributo CodiceProdotto della tabella COMBATTUTEDA e l'attributo CodiceProdotto della tabella PRODOTTICHIMICI
- Esiste un vincolo di integrità referenziale tra l'attributo AgentePatogeno della tabella COMBATTUTEDA e l'attributo AgentePatogeno della tabella PATOLOGIE
- Esiste un vincolo di integrità referenziale tra l'attributo codiceProdotto della tabella TRATTAMENTO e l'attributo codiceProdotto della tabella PRODOTTICHIMICI
- Esiste un vincolo di integrità referenziale tra l'attributo codiceEsemplare della tabella TRATTAMENTO e l'attributo codiceEsemplare della tabella ESEMPLARE
- Esiste un vincolo di integrità referenziale tra l'attributo idTerreno della tabella COMPOSTODA e l'attributo idTerreno della tabella TERRENO
- Esiste un vincolo di integrità referenziale tra l'attributo Codice della tabella COMPOSTODA e l'attributo Codice della tabella COMPONENTI
- Esiste un vincolo di integrità referenziale tra l'attributo CodicePianta della tabella CONCIMAZIONE e l'attributo CodicePianta della tabella PIANTA
- Esiste un vincolo di integrità referenziale tra l'attributo idSostanza della tabella CONCIMAZIONE e l'attributo CodicePianta della tabella SOSTANZE
- Esiste un vincolo di integrità referenziale tra l'attributo CodiceRipiano della tabella CONTENITORE e l'attributo CodiceRipiano della tabella RIPIANO
- Esiste un vincolo di integrità referenziale tra l'attributo idTerreno della tabella CONTIENE1 e l'attributo idTerreno della tabella TERRENO
- Esiste un vincolo di integrità referenziale tra l'attributo CodiceElemento della tabella CONTIENE1 e l'attributo Sigla della tabella ELEMENTI
- Esiste un vincolo di integrità referenziale tra l'attributo idContenitore della tabella CONTIENE2 e l'attributo idContenitore della tabella CONTENITORE
- Esiste un vincolo di integrità referenziale tra l'attributo idTerreno della tabella CONTIENE2 e l'attributo idTerreno della tabella TERRENO
- Esiste un vincolo di integrità referenziale tra l'attributo CodicePianta della tabella ESEMPLARE e l'attributo CodicePianta della tabella PIANTA
- Esiste un vincolo di integrità referenziale tra l'attributo idContenitore della tabella ESEMPLARE e l'attributo idContenitore della tabella CONTENITORE
- Esiste un vincolo di integrità referenziale tra l'attributo id della tabella ESIGENZE e l'attributo id della tabella PERIODO
- Esiste un vincolo di integrità referenziale tra l'attributo CodiceSintomo della tabella FOTO e l'attributo CodiceSintomo della tabella SINTOMI
- Esiste un vincolo di integrità referenziale tra l'attributo file della tabella FOTO e l'attributo file della tabella IMMAGINE
- Esiste un vincolo di integrità referenziale tra l'attributo CodiceSerra della tabella INFOAMBIENTE e l'attributo CodiceSerra della tabella SERRA
- Esiste un vincolo di integrità referenziale tra l'attributo idContenitore della tabella INFOSALUTE e l'attributo idContenitore della tabella CONTENITORE

- Esiste un vincolo di integrità referenziale tra l'attributo CodiceEsemplare della tabella INTERVENTO e l'attributo CodiceEsemplare della tabella ESEMPLARE
- Esiste un vincolo di integrità referenziale tra l'attributo codicePianta della tabella MANUTENZIONE e l'attributo codicePianta della tabella PIANTA
- Esiste un vincolo di integrità referenziale tra l'attributo AgentePatogeno della tabella MALATTIA e l'attributo AgentePatogeno della tabella PATOLOGIE
- Esiste un vincolo di integrità referenziale tra l'attributo codiceEsemplare della tabella MALATTIA e l'attributo codiceEsemplare della tabella ESEMPLARE
- Esiste un vincolo di integrità referenziale tra l'attributo username della tabella INTERVENTO e l'attributo username della tabella UTENTE
- Esiste un vincolo di integrità referenziale tra l'attributo username della tabella NOTIFICHE e l'attributo username della tabella UTENTE
- Esiste un vincolo di integrità referenziale tra l'attributo username della tabella ORDINE e l'attributo username della tabella UTENTE
- Esiste un vincolo di integrità referenziale tra l'attributo CodiceEsemplare della tabella ORDINE e l'attributo CodiceEsemplare della tabella ESEMPLARE
- Esiste un vincolo di integrità referenziale tra l'attributo CodicePianta della tabella PERIDO e l'attributo CodicePianta della tabella PIANTA
- Esiste un vincolo di integrità referenziale tra l'attributo idTerreno della tabella PIANTA e l'attributo idTerreno della tabella TERRENO
- Esiste un vincolo di integrità referenziale tra l'attributo CodiceSettore della tabella POSIZIONEPIANTE e l'attributo CodiceSettore della tabella SETTORE
- Esiste un vincolo di integrità referenziale tra l'attributo codicePianta della tabella POSIZIONEPIANTE e l'attributo codicePianta della tabella PIANTA
- Esiste un vincolo di integrità referenziale tra l'attributo username della tabella POST e l'attributo username della tabella UTENTE
- Esiste un vincolo di integrità referenziale tra l'attributo username della tabella POSTRISPOSTA e l'attributo username della tabella UTENTE
- Esiste un vincolo di integrità referenziale tra l'attributo codicePost della tabella POSTRISPOSTA e l'attributo codicePost della tabella POST
- Esiste un vincolo di integrità referenziale tra l'attributo CodiceSezione della tabella SEZIONI e l'attributo CodiceSezione della tabella RIPIANO
- Esiste un vincolo di integrità referenziale tra l'attributo CodiceSerra della tabella SERRA e l'attributo CodiceSerra della tabella RIPIANO
- Esiste un vincolo di integrità referenziale tra l'attributo CodiceSerra della tabella SERRA e l'attributo CodiceSerra della tabella SEZIONI
- Esiste un vincolo di integrità referenziale tra l'attributo CodiceSettore della tabella SEGMENTO e l'attributo CodiceSettore della tabella SETTORE
- Esiste un vincolo di integrità referenziale tra l'attributo CodiceSede della tabella SERRA e l'attributo CodiceSede della tabella SEDE
- Esiste un vincolo di integrità referenziale tra l'attributo username della tabella SPAZIOVERDE e l'attributo username della tabella UTENTE
- Esiste un vincolo di integrità referenziale tra l'attributo username della tabella VALUTAZIONE e l'attributo username della tabella UTENTE
- Esiste un vincolo di integrità referenziale tra l'attributo Codice della tabella VALUTAZIONE e l'attributo Codice della tabella POSTRISPOSTA

## 2.8. Normalizzazione

Elenco delle dipendenze funzionali.

### Tabella PIANTA:

PIANTA (codicePianta, idTerreno, cultivar, nome, sempreverde, dioica, indiceAccrAereo, indiceAccrRadici, indiceManutenzione, infestante)

- codicePianta, idTerreno → cultivar, nome, sempreverde, dioica, indiceAccrAereo, indiceAccrRadici, indiceManutenzione, infestante

Poiché la parte sinistra è superchiave, PIANTA è in BCNF.

### Tabella PERIODO:

PERIODO (id, codicePianta, tipoPeriodo, meseInizio, meseFine)

- id, codicePianta → tipoPeriodo, meseInizio, meseFine

Poiché la parte sinistra è superchiave, PERIODO è in BCNF.

### Tabella ESIGENZE:

ESIGENZE (id, tipoLuce, oreLuce, tempMin, tempMax, quantitaAcqua, periodicitAcqua)

- id → tipoLuce, oreLuce, tempMin, tempMax, quantitaAcqua, periodicitAcqua

Poiché la parte sinistra è superchiave, ESIGENZE è in BCNF.

### Tabella CONCIMAZIONE:

CONCIMAZIONE (idSostanza, codicePianta, periodicitAcqua, quantita, somministrazione, tipoPeriodo, meseInizio, meseFine)

- idSostanza, codicePianta → periodicitAcqua, quantita, somministrazione, tipoPeriodo, meseInizio, meseFine

Poiché la parte sinistra è superchiave, CONCIMAZIONE è in BCNF.

### Tabella SOSTANZE:

SOSTANZE (idSostanza, nome)

- idSostanza → nome

Poiché la parte sinistra è superchiave, SOSTANZE è in BCNF.

### Tabella TERRENO:

TERRENO (idTerreno, permeabilita, PH)

- idTerreno → permeabilita, PH

Poiché la parte sinistra è superchiave, TERRENO è in BCNF.

### Tabella CONTINE1:

CONTIENE1 (idTerreno, sigla, concentrazione)

- idTerreno, sigla → concentrazione

Poiché la parte sinistra è superchiave, CONTIENE1 è in BCNF.

### Tabella CONTINE2:

CONTIENE2 (idTerreno, idContenitore)

Poiché non esistono dipendenze funzionali, CONTIENE2 è in BCNF.

### Tabella COMPOSTODA:

COMPOSTODA (idTerreno, codice, concentrazione)

- idTerreno, codice → concentrazione

Poiché la parte sinistra è superchiave, COMPOSTODA è in BCNF.

**Tabella ELEMENTI:**

ELEMENTI (sigla, nome)

- sigla → nome

Poiché la parte sinistra è superchiave, ELEMENTI è in BCNF.

**Tabella COMPONENTI:**

COMPONENTI (codice, nome)

- codice → nome

Poiché la parte sinistra è superchiave, COMPONENTI è in BCNF.

**Tabella ESEMPLARE:**

ESEMPLARE (codiceEsemplare, idContenitore, codicePianta, prezzo, dimParteRad)

- codiceEsemplare, idContenitore, codicePianta → nome, dimParteRad

Poiché la parte sinistra è superchiave, ESEMPLARE è in BCNF.

**Tabella ORDINE:**

ORDINE (username, codiceEsemplare, dataAcquisto, codiceOrdine, stato)

- username, codiceEsemplare → dataAcquisto, codiceOrdine, stato

Poiché la parte sinistra è superchiave, ORDINE è in BCNF.

**Tabella UTENTE:**

UTENTE (username, password, nome, cognome, email, domandaSegreta, rispostaSegreta, citta)

- username → password, nome, cognome, email, domandaSegreta, rispostaSegreta, citta

Poiché la parte sinistra è superchiave, UTENTE è in BCNF.

**Tabella NOTIFICA:**

NOTIFICA (codiceNotifica, username, testo, data)

- codiceNotifica, username → testo, data

Poiché la parte sinistra è superchiave, NOTIFICA è in BCNF.

**Tabella POST:**

POST (codicePost, username, titolo, testo, forum, dataPubblicazione, link)

- codicePost, username → titolo, testo, forum, dataPubblicazione, link

Poiché la parte sinistra è superchiave, POST è in BCNF.

**Tabella POSTRISPOSTA:**

POSTRISPOSTA (codiceRisposta, username, codicePost, data, testo)

- codiceRisposta, username, codicePost → data, testo

Poiché la parte sinistra è superchiave, POSTRISPOSTA è in BCNF.

**Tabella VALUTAZIONE:**

VALUTAZIONE (codiceRisposta, username, stelle)

- codiceRisposta, username → stelle

Poiché la parte sinistra è superchiave, VALUTAZIONE è in BCNF.

**Tabella INTERVENTO:**

MANUTENZIONE (idIntervento, username, codiceEsemplare, tipoRichiesta, scadenza, costo, frequenza, tipologia, descrizione, stato)

- idIntervento, username → codiceEsemplare, tipoRichiesta, scadenza, costo, frequenza, tipologia, descrizione, stato

Poiché la parte sinistra è superchiave, MANUTENZIONE è in BCNF.

**Tabella MANUTENZIONE:**

INTERVENTO (idManutenzione, codicePianta, tipologia, descrizione, periodo)

- idManutenzione, codicePianta → tipologia, descrizione, periodo

Poiché la parte sinistra è superchiave, INTERVENTO è in BCNF.

**Tabella SPAZIOVERDE:**

SPAZIOVERDE (id, username, area)

- id, username → area

Poiché la parte sinistra è superchiave, SPAZIOVERDE è in BCNF.

**Tabella SETTORE:**

SETTORE (codiceSettore, idSpazio, tipoTerreno, oreLuce, puntoCardinale)

- codiceSettore, idSpazio → tipoTerreno, oreLuce, puntoCardinale

Poiché la parte sinistra è superchiave, SETTORE è in BCNF.

**Tabella SEGMENTO:**

SEGMENTO (codiceSettore, idSpazio, Xiniziale, Yiniziale, Xfinale, Yfinale)

- codiceSettore, idSpazio → Xiniziale, Yiniziale, Xfinale, Yfinale

Poiché la parte sinistra è superchiave, SEGMENTO è in BCNF.

**Tabella POSIZIONEPIANTE:**

POSIZIONE PIANTE (codiceSettore, codicePianta, posX, posY)

Poiché non esistono dipendenze funzionali, POSIZIONEPIANTE è in BCNF.

**Tabella MALATTIA:**

MALATTIA (agentePatogeno, codiceEsemplare, data)

- agentePatogeno, codiceEsemplare → data

Poiché la parte a sinistra è superchiave, MALATTIA è in BCNF.

**Tabella PATOLOGIE:**

PATOLOGIE (agentePatogeno)

Poiché non esistono dipendenze funzionali, PATOLOGIE è in BCNF.

**Tabella CARATTERIZZATEDA:**

CARATTERIZZATEDA (codiceSintomo, agentePatogeno)

Poiché non esistono dipendenze funzionali, CARATTERIZZATEDA è in BCNF.

**Tabella SINTOMI:**

SINTOMI (codiceSintomo, descrizione)

- codiceSintomo → descrizione

Poiché la parte sinistra è superchiave, SINTOMI è in BCNF.

**Tabella FOTO:**

FOTO (codiceSintomo, file)

Poiché non esistono dipendenze funzionali, FOTO è in BCNF.

**Tabella IMMAGINE:**

IMMAGINE (file, data, dimensioni)

- file → data, dimensioni

Poiché la parte sinistra è superchiave, IMMAGINE è in BCNF.

**Tabella COMBATTUTEDA:**

COMBATTUTEDA (agentePatogeno, codiceProdotto, dosaggio)

- agentePatogeno, codiceProdotte → dosaggio

Poiché la parte a sinistra è superchiave, COMBATTUTEDA è in BCNF.

**Tabella PRODOTTICHIMICI:**

PRODOTTI CHIMICI (codiceProdotto, periodoVietato, tempoAttesa, principioAttivo, tipologia, somministrazione)

- codiceProdotto → periodoVietato, tempoAttesa, principioAttivo, tipologia, somministrazione

Poiché la parte sinistra è superchiave, PRODOTTICHIMICI è in BCNF.

**Tabella TRATTAMENTO:**

TRATTAMENTO (codiceProdotto, codiceEsemplare, codiceTrattamento, dose, data)

- codiceProdotto, codiceEsemplare → codiceTrattamento, dose, data

Poiché la parte sinistra è superchiave, TRATTAMENTO è in BCNF.

**Tabella CONTENITORE:**

CONTENITORE (idContenitore, codiceRipiano, superficie)

- idContenitore, codiceRipiano → superficie

Poiché la parte sinistra è superchiave, CONTENITORE è in BCNF.

**Tabella INFOSALUTE:**

INFOSALUTE (idContenitore, data, ora, umidita, irrigazione)

- idContenitore, data, ora → umidita, irrigazione

Poiché la parte sinistra è superchiave, INFOSALUTE è in BCNF.

**Tabella RIPIANO:**

RIPIANO (codiceRipiano, codiceSezione, codiceSerra, irrigazioneRip)

- codiceRipiano, codiceSezione, codiceSerra → irrigazioneRip

Poiché la parte sinistra è superchiave, RIPIANO è in BCNF.

**Tabella SEZIONI:**

SEZIONI (codiceSezione, codiceSerra, nomeSezione, riempimento, capienza, temperatura, luce, umidita, irrigazione)

- codiceSezione, codiceSerra → nomeSezione, riempimento, capienza, temperatura, luce, umidità, irrigazione

Poiché la parte sinistra è superchiave, SEZIONI è in BCNF.

#### **Tabella SERRA:**

SERRA (codiceSerra, codiceSede, nome, indirizzo, altezza, larghezza, lunghezza, numMaxPiante, numMinPiante)

- codiceSerra, codiceSede → nome, indirizzo, altezza, larghezza, lunghezza, numMaxPiante, numMinPiante

Poiché la parte sinistra è superchiave, SERRA è in BCNF.

#### **Tabella INFOAMBIENTE:**

INFOAMBIENTE (codiceSerra, data, ora, umidità, temperatura, luce, irrigazione)

- codiceSerra, data, ora → umidità, temperatura, luce, irrigazione

Poiché la parte sinistra è superchiave, INFOAMBIENTE è in BCNF.

#### **Tabella SEDE:**

SEDE (codiceSede, nome, indirizzoSede, numDipendenti)

- codiceSede → nome, indirizzoSede, numDipendenti

Poiché la parte sinistra è superchiave, SEDE è in BCNF.

### 3. Trigger ed Eventi

**-Evento per la compilazione dei report sui dati ambientali di una serra. Compilato ogni giorno, per ogni serra:**

```
DROP EVENT IF EXISTS CompilaReport;
DELIMITER $$
CREATE EVENT CompilaReport
ON SCHEDULE EVERY 1 DAY
STARTS '2017-02-08 23:55:00'
DO
BEGIN
    INSERT INTO reportSerra
    SELECT codiceSerra,
           CURRENT_DATE,
           AVG(umidita),
           AVG(luce),
           AVG(irrigazione),
           AVG(temperatura)
    FROM infoambiente
    WHERE data=CURRENT_DATE
    GROUP BY codiceSerra;
END $$
DELIMITER ;
```



**-Trigger che impedisce di valutarci un post da soli:**

```
DROP TRIGGER IF EXISTS BloccaValutazione;

DELIMITER $$

CREATE TRIGGER BloccaValutazione BEFORE INSERT ON valutazione
FOR EACH ROW
BEGIN
    DECLARE _control INTEGER DEFAULT 0;

    IF EXISTS (
        SELECT username, codiceRisposta
        FROM postRisposta
        WHERE username=NEW.username AND codiceRisposta=NEW.codiceRisposta
    ) THEN SET _control=1;
END IF;

IF (_control = 1) THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Non puoi valutarti un post da solo!';
END IF;
END $$
DELIMITER ;
```

**-Trigger che mantiene aggiornato l'attributo numPost della tabella Utente:**

```
DROP TRIGGER IF EXISTS contaPost;

DELIMITER $$
CREATE TRIGGER contaPost
AFTER INSERT ON post FOR EACH ROW
BEGIN
    UPDATE utente U
    SET U.NumPost=U.NumPost+1
    WHERE U.username=NEW.username;

END$$
DELIMITER ;
```

*(stessa cosa per la tabella PostRisposta)*

**-Trigger che mantiene aggiornato l'attributo numAcquisti della tabella Pianta:**

```
DROP TRIGGER IF EXISTS aggiornaAcquisti;

DELIMITER $$
CREATE TRIGGER aggiornaAcquisti
AFTER UPDATE ON ordine FOR EACH ROW
BEGIN
    IF(NEW.stato='evaso') THEN
        UPDATE pianta P
        SET P.numAcquisti = P.numAcquisti + 1
        WHERE P.codice = (
            SELECT E.codicePianta
            FROM esemplare E
            WHERE E.codiceEsemplare=NEW.codiceEsemplare
        );
    END IF;
END$$
DELIMITER ;
```

*Esiste un altro trigger identico a questo sull'INSERT nella medesima tabella. Questo perché se un cliente acquista di persona una pianta presso una delle serre, questo acquisto viene registrato direttamente come un ordine evaso.*

**-Trigger che controlla che lo stato di un ordine sia consistente con il flusso previsto:**

```
DROP TRIGGER IF EXISTS controlloFlussoOrdine;
DELIMITER $$
CREATE TRIGGER controlloFlussoOrdine
BEFORE UPDATE ON ordine FOR EACH ROW
BEGIN
    DECLARE _statoPrec CHAR(50);
    SET _statoPrec = (
        SELECT stato
        FROM ordine
        WHERE codiceOrdine = NEW.codiceOrdine);

    IF((_statoPrec = 'processazione' AND NEW.stato<>'preparazione')
        OR (_statoPrec = 'preparazione' AND NEW.stato<>'spedito')
        OR (_statoPrec = 'spedito' AND NEW.stato<>'evaso')) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT= 'ERRORE: Valore stato ordine non consentito.';
    END IF;
END$$
DELIMITER ;
```

**-Evento che compila un report settimanale relativamente agli ordini pendenti:**

```
DROP EVENT IF EXISTS ordiniPendenti;
DELIMITER $$
CREATE EVENT ordiniPendenti
ON SCHEDULE EVERY 7 DAY
STARTS '2017-02-08 23:55:00'
DO
BEGIN
    DELETE FROM reportOrdiniPendenti;

    INSERT INTO reportOrdiniPendenti
    SELECT E.codicePianta, COUNT(*) AS NumeroPendenti
    FROM ordine O
    NATURAL JOIN
    Esemplare E
    WHERE O.stato = 'pendente'
    GROUP BY E.codicePianta;
END $$
DELIMITER ;
```

**-Function per il calcolo dell'Indice di credibilità di un utente:**

```
DROP FUNCTION IF EXISTS calcolaIndiceCredibilita;
DELIMITER $$
CREATE FUNCTION calcolaIndiceCredibilita(_utente CHAR(50))
```

```

RETURNS CHAR(20) NOT DETERMINISTIC
BEGIN
    DECLARE _credibilita CHAR(20) DEFAULT '0.00%';
    DECLARE _maxNumStelle INT DEFAULT 0;
    DECLARE _numeroStelle INT DEFAULT 0;
    DECLARE _numeroPost INT DEFAULT 0;
    DECLARE _totalePost INT DEFAULT 0;
    DECLARE _mediaPost DOUBLE DEFAULT 0;
    DECLARE _mediaStelle DOUBLE DEFAULT 0;

    SET _maxNumStelle = 5*(
        SELECT IFNULL(COUNT(*),0)
        FROM valutazione V INNER JOIN postRisposta PR
            ON V.codiceRisposta=PR.codiceRisposta
        WHERE PR.username=_utente
    );

    SET _numeroStelle = (
        SELECT IFNULL(SUM(V.stelle),0)
        FROM valutazione V INNER JOIN postrisposta P
            ON V.codiceRisposta=P.codiceRisposta
        WHERE P.username = _utente
    );

    SET _numeroPost = (
        SELECT U.numPost
        FROM utente U
        WHERE U.username=_utente
    );

    SET _totalePost = (
        SELECT SUM(D.numero)
        FROM (
            SELECT COUNT(*) AS numero
            FROM post
            UNION
            SELECT COUNT(*) AS numero
            FROM postRisposta
        ) AS D);

    IF(_totalePost>0) THEN
        SET _mediaPost = ROUND((_numeroPost*100)/_totalePost,2);
    END IF;

    IF(_maxNumStelle>0) THEN
        SET _mediaStelle = ROUND((_numeroStelle*100)/_maxNumStelle,2);
    END IF;

```

```

    SET _credibilita = CONCAT((_mediaPost+_mediaStelle)/2,'%');
    RETURN (_credibilita);
END$$
DELIMITER ;

```

**-Evento per la compilazione giornaliera del Report Diagnostica:**

```

DROP EVENT IF EXISTS compilaReportDiagnostica;

DELIMITER $$
CREATE EVENT compilaReportDiagnostica
ON SCHEDULE EVERY 1 DAY
STARTS '2017-02-09 23:55:00'
DO
BEGIN
    DELETE FROM reportDiagnostica;

    INSERT INTO reportDiagnostica
    SELECT CURRENT_DATE, M.codiceEsemplare, S.descrizione, M.agentePatogeno
    FROM Malattia M
        NATURAL JOIN
        caratterizzata da CD
        NATURAL JOIN
        sintomi S
    GROUP BY M.codiceEsemplare, M.agentePatogeno;

END $$
DELIMITER ;

```

**-Trigger che libera un contenitore una volta che un esemplare è stato venduto:**

```
DROP TRIGGER IF EXISTS aggiornaContenitori;
DELIMITER $$

CREATE TRIGGER aggiornaContenitori
AFTER UPDATE ON ordine FOR EACH ROW
BEGIN
    IF(NEW.stato='spedito') THEN
        UPDATE esemplare E
        SET E.idContenitore = NULL
        WHERE E.codiceEsemplare = NEW.codiceEsemplare;
    END IF;
END$$
DELIMITER ;
```

## 4. Area Analytics

### 4.1. Smart Design

La funzionalità di smart design permette all'utente di riprogettare un settore del giardino scegliendo le piante in base al costo e all'indice di manutenzione; inoltre si vuole far visualizzare all'utente anche i vari periodi che ciascuna pianta attraversa affinché possa scegliere piante che fioriscono in diverse stagioni dell'anno.

```
DROP PROCEDURE IF EXISTS smartDesign;

DELIMITER $$
CREATE PROCEDURE smartDesign(IN _indiceManutenzione CHAR(5),
                             IN _costo INT(11))
BEGIN
    DECLARE indManutenzione INTEGER DEFAULT 0;

    IF(_indiceManutenzione <>'basso' AND _indiceManutenzione <>'medio' AND _indiceManutenzione
    <> 'alto')
    THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Indice manutenzione non valido.';
    END IF;

    IF _indiceManutenzione = 'basso' THEN
        SET indManutenzione = 2;
    ELSEIF _indiceManutenzione = 'medio' THEN
        SET indManutenzione = 4;
    ELSE SET indManutenzione = 5;
    END IF;

    CREATE OR REPLACE VIEW costoMedioPiante AS
    SELECT codicePianta, ROUND(AVG(prezzo),2) AS prezzo
    FROM esemplare
    GROUP BY codicePianta;

    SELECT codicePianta, prezzo, tipoPeriodo, meseInizio, meseFine
    FROM Pianta NATURAL JOIN Periodo NATURAL JOIN costoMedioPiante
    WHERE indiceManutenzione <= indManutenzione
        AND prezzo <= _costo
    GROUP BY codicePianta, tipoPeriodo;

END $$
DELIMITER ;
```

Quindi, la procedura prende in ingresso l'indice di manutenzione e il costo selezionati dall'utente e come prima cosa verifica che l'indice assuma valori tra [basso, medio, alto]. Con la view *costoMedioPiante* si proietta, per ogni pianta, il costo medio in base agli esemplari di quella specie presenti in serra. La selezione finale fornisce una lista di piante coincidenti con le richieste dell'utente.

Un esempio di chiamata è il seguente:

```
CALL smartDesign('medio', 20);
```

## 4.2. Reporting

Come funzionalità di reporting vogliamo avere la possibilità di individuare quali sono le patologie/attacchi parassitari che più colpiscono una determinata specie di piante oppure sapere quali sono le specie meno vendute.

### Sensibilità alle patologie

La prima informazione possiamo recuperarla andando a contare quante volte una patologia colpisce una determinata pianta e proiettando la patologia con il maggior numero di casi per quella specie. Nello specifico, tale funzionalità può essere implementata mediante la seguente *stored procedure*:

```
DROP PROCEDURE IF EXISTS sensibilitaPatologie;
DELIMITER $$
CREATE PROCEDURE sensibilitaPatologie(IN _specie CHAR(50))
BEGIN
    CREATE OR REPLACE VIEW casiPatologie AS
    SELECT P.nome AS Specie, M.agentePatogeno, COUNT(*) AS NumCasi
    FROM malattia M NATURAL JOIN esemplare E NATURAL JOIN pianta P
    GROUP BY M.agentePatogeno, P.nome
    ORDER BY Specie;

    SELECT *
    FROM casiPatologie C
    WHERE C.Specie=_specie
    GROUP BY C.Specie
    HAVING C.NumCasi = (
        SELECT MAX(C2.NumCasi)
        FROM casiPatologie C2
        WHERE C2.Specie=C.Specie
    );
END$$
DELIMITER ;
```

La procedura *sensibilitaPatologie* prende in ingresso il nome di una determinata specie e produce come result set la patologia (oppure le patologie) che maggiormente ha colpito quella specie, e il numero di casi.

Un esempio di chiamata è il seguente:

```
CALL sensibilitaPatologie('Abete');
```



### Analisi delle vendite

Analizzando le vendite è possibile scoprire quali specie sono le meno vendute e quali sono le più vendute. Con queste informazioni, i dipendenti dell'azienda possono decidere, se la domanda è alta, di ordinare più esemplari di una determinata specie oppure di bloccare gli ordini relativi alla specie meno venduta.

Queste informazioni le possiamo visualizzare come il tasso di vendita di una determinata specie. Andiamo cioè a calcolare la percentuale di esemplari venduti rispetto al totale che erano stati ordinati dall'azienda.

Si può ipotizzare che una specie risulti poco venduta quando questo tasso sia inferiore al 20% e che una specie sia molto richiesta se il tasso risulta superiore all'80%.

Nello specifico, questa funzionalità la implementiamo mediante la seguente *stored function*:

```
DROP FUNCTION IF EXISTS analizzaVendite;
DELIMITER $$
CREATE FUNCTION analizzaVendite(_codice CHAR(50))
RETURNS CHAR(11) NOT DETERMINISTIC
BEGIN
    DECLARE _vendite CHAR(11) DEFAULT '0.00%';
    DECLARE _quantitaMagazzino INT(11) DEFAULT 0;
    DECLARE _quantitaVendite INT(11) DEFAULT 0;
    DECLARE _quantitaTotale INT(11) DEFAULT 0;
    DECLARE _control TINYINT DEFAULT 0;

    IF( (SELECT nome FROM pianta WHERE codicePianta=_codice) IS NULL) THEN
        SET _control=1;
    END IF;

    IF(_control=1) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT='Codice Pianta non valido.';
    END IF;

    SET _quantitaMagazzino = (
        SELECT COUNT(*) AS Quantita
        FROM esemplare E NATURAL JOIN pianta P
        WHERE idContenitore IS NOT NULL AND P.codicePianta=_codice
        GROUP BY P.Nome);

    SET _quantitaVendite = (
        SELECT COUNT(*) AS QuantitaVendite
        FROM ordine O NATURAL JOIN esemplare E NATURAL JOIN pianta P
        WHERE P.codicePianta=_codice
        GROUP BY P.nome);

    SET _quantitaTotale = _quantitaVendite + _quantitaMagazzino;
    SET _vendite = CONCAT(ROUND((_quantitaVendite*100)/(_quantitaTotale),2),'%');
```

```

IF(_vendite IS NULL) THEN
    SET _vendite = '0.00%';
END IF;

RETURN(_vendite);
END$$
DELIMITER ;

```

La funzione prende in ingresso il codice di un esemplare e verifica innanzitutto che sia un codice valido. Dopodiché procede al calcolo del tasso di vendita.

Un esempio di chiamata è il seguente:

```
SELECT '000' AS codicePianta, analizzaVendite('000') AS TassoVendite
```

Per quanto riguarda il caso illustrato, l'interrogazione produrrà il seguente *result set*:

codicePianta	TassoVendite
000	60.00%

### 4.3. Indagini Statistiche

Data una patologia, vogliamo analizzare i dati memorizzati nel database per fornire ad un esperto le informazioni ambientali delle serre. L'esperto potrà poi verificare se ci sono anomalie in queste informazioni ambientali e valutare se c'è una correlazione tra l'infestazione e le condizioni ambientali delle serre.

Abbiamo pensato di implementare questa funzionalità mediante la seguente *stored procedure*:

```

DROP PROCEDURE IF EXISTS analisiPatologie;

DELIMITER $$
CREATE PROCEDURE analisiPatologie(IN _patologia CHAR(50))
BEGIN
    CREATE OR REPLACE VIEW prospetto AS
    SELECT M.*, E.codicePianta, R.codiceSerra
    FROM malattia M NATURAL JOIN esemplare E NATURAL JOIN contenitore C NATURAL JOIN
    ripiano R
    WHERE E.idContenitore IS NOT NULL;

    SELECT P.agentePatogeno, P.codiceEsemplare, P.codicePianta, I.codiceSerra, AVG(I.umidita)
    AS umiditaMedia, AVG(I.luce) AS luceMedia, AVG(I.irrigazione) AS irrigazioneMedia,
    AVG(I.temperatura) AS temperaturaMedia
    FROM infoambiente I INNER JOIN prospetto P
    ON I.codiceSerra=P.codiceSerra
    WHERE I.data >= (P.data - INTERVAL 1 MONTH) AND P.agentePatogeno=_patologia
    GROUP BY P.agentePatogeno, P.codiceEsemplare, P.codicePianta, I.codiceSerra;
END$$
DELIMITER ;

```

Quest'ultima prende in ingresso una patologia e, per ogni esemplare colpito da essa, va a recuperare le condizioni ambientali della serra nella quale era riposto, relativamente all'ultimo mese.

Un esempio di chiamata è il seguente:

```
CALL analisiPatologie('punteruolo rosso');
```