



Progetto di Basi di Dati

Marco Micera

Anno accademico 2015-2016

Indice

1. Analisi delle specifiche	<i>pagina 6</i>
2. Progettazione concettuale	<i>pagina 12</i>
3. Ristrutturazione del diagramma E-R	<i>pagina 19</i>
4. Individuazione di operazioni sui dati	<i>pagina 24</i>
5. Analisi delle prestazioni delle operazioni	<i>pagina 25</i>
6. Introduzione di ridondanze	<i>pagina 41</i>
7. Progettazione logica	<i>pagina 54</i>
8. Analisi delle dipendenze funzionali e normalizzazione	<i>pagina 59</i>
9. Implementazione su DBMS Oracle MySQL	<i>pagina 65</i>

1. Analisi delle specifiche

1.1 Assunzioni

- Le esigenze di luce in termini di posizione (pieno sole, mezz'ombra e ombra) non variano dal periodo vegetativo a quello di riposo.
- Una pianta in magazzino risiede in un contenitore. Una pianta venduta è collocata in un vaso o in piena terra.
- Un contenitore del magazzino può contenere al massimo una pianta.

1.2 Glossario dei termini

Termine	Descrizione	Sinonimi	Collegamenti
Pianta	Prodotto acquistabile dall'utente presso l'impresa.	Esemplare	Terreno, Elemento, Profilo, Manutenzione
Terreno	Dove la pianta viene messa a dimora, sia in vaso che in piena terra.	Sottosuolo, substrato	Elemento, Pianta, Componenti
Elemento	Presente nel terreno, influisce sullo stato generale della pianta.	Sostanza	Pianta, Terreno
Componenti	Determinano la consistenza e la permeabilità di un terreno.		Terreno
Manutenzione	Operazione (potatura, rinvaso o concimazione) da effettuare sulla pianta.	Intervento di manutenzione, richiesta	Pianta
Cliente	Persona fisica in possesso di un account	Utente	Profilo
Profilo	Collegati ad accounts, tiene traccia delle piante a cui un utente è appassionato e delle schede delle piante acquistate	Area riservata	Cliente, Pianta, Arredamento
Spazio	Giardino che il cliente ha a disposizione per la disposizione delle piante; suddivisibile in settori.	Spazio verde, giardino	Arredamento
Arredamento	Insieme di settori e piante (invasate o meno) di uno spazio	Prospetto, architettura del giardino	Spazio, Profilo

1.3 Strutturazione dei requisiti in gruppi di frasi omogenee

Frasi di carattere generale
Il database che si desidera progettare è volto a contenere i dati a supporto delle funzionalità del sistema informativo di una grande impresa distribuita sul territorio nazionale che si occupa di giardinaggio. L'impresa è dislocata in diverse sedi, nelle quali ospita svariati tipi di piante per la vendita online. Tali piante sono gestite secondo principi di smart gardening. In particolare, fra le altre funzionalità, un sistema intelligente provvede ai fabbisogni delle piante e a riconoscere istantaneamente le patologie e le carenze di elementi, per poter somministrare il giusto principio chimico nel più breve tempo possibile. Per fare ciò, l'azienda si avvale di droni muniti di microcamere che riescono a individuare visivamente le principali patologie, rappresentando un valido ausilio per la diagnosi. Le eventuali carenze idriche e/o di macro/micro elementi sono rilevate mediante sensori presenti nel terreno. I trattamenti contro le patologie rilevate

sono effettuati in automatico e il personale è allertato ogniqualvolta il sistema automatico rileva anomalie che non sono gestibili direttamente. 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, 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. Un'altra attività dell'azienda consiste nella manutenzione degli spazi verdi. La manutenzione è gestita mediante richiesta diretta del cliente, oppure in modo automatico. Particolari funzionalità di notifica permettono di segnalare eventuali necessità di interventi di potatura, concimazione, o diverso intervento, direttamente sullo smartphone (o smartwatch) del cliente. In ottica di manutenzione, l'azienda effettua rinvasi, piantumazioni, trattamenti contro patologie, concimazioni e potature. L'azienda ha un sito web che ospita una piattaforma social mediante la quale possono interagire, come sarà descritto in dettaglio nel seguito. Inoltre, il sito web ospita lo store tramite il quale i clienti possono acquistare esemplari di piante per il giardino, nonché prenotare interventi di manutenzione e/o effettuare la progettazione dei propri spazi verdi.

Fraasi relative alle piante

Una pianta è caratterizzata da un nome, un genere, una cultivar, una dimensione massima raggiungibile (in termini di volume) e un indice di accrescimento della parte aerea e della parte radicale. L'indice di accrescimento è un indice che dà informazioni circa la velocità con cui la pianta giunge alla dimensione massima raggiungibile. Una pianta è poi caratterizzata da uno o più periodi di fioritura e/o fruttificazione e da un insieme di esigenze di luce, temperatura, terreno, irrigazione e concimazione. Inoltre, una pianta possiede un periodo vegetativo e uno di riposo che si alternano durante l'anno. Infine, una pianta può essere sempreverde o a foglie caduche, e può o meno essere dioica, cioè vi sono esemplari maschili e femminili della stessa.

In base alle sue caratteristiche ed esigenze, una pianta è associata a un indice di manutenzione. L'indice di manutenzione dà informazioni circa il livello di cure che una pianta richiede nel tempo. L'indice di manutenzione tiene in considerazione anche l'indice di accrescimento. Relativamente alle esigenze di illuminazione, una pianta può richiedere il pieno Sole, la mezz'ombra o l'ombra. Una pianta può richiedere, inoltre, un numero minimo di ore di luce al giorno. Il numero di ore di luce al giorno varia in base al periodo in cui si trova la pianta: vegetativo o riposo. La luce può essere sia diretta che indiretta, dipendentemente dal fatto che la pianta possa, rispettivamente, essere esposta direttamente o meno ai raggi del Sole. Circa la temperatura, ogni genere di pianta ha un range di temperature nel quale può vivere. In particolare, spesso la temperatura minima è il valore più importante perché al di sotto di essa la linfa congela e la pianta perde la parte aerea oppure muore. [...] Oltre che dall'approvvigionamento di elementi, il conflitto può anche essere originato dalle particolari caratteristiche dell'apparato radicale o della parte aerea di un genere di pianta. In particolare, per ciascuna pianta che può cagionare conflitti, è stabilita una distanza minima più vicino della quale è bene non mettere a dimora altre piante perché, nel tempo, le due piante entreranno in conflitto. Relativamente all'apparato radicale, non possono essere messe a dimora troppo in prossimità piante che necessitano degli stessi elementi perché prima o poi finiranno per sottrarsi. In merito alla parte aerea, a causa dell'espansione della stessa, è possibile che si generino sofferenze dovute alla mancanza di luce. I conflitti relativi alla parte aerea possono essere stimati in base all'indice di accrescimento. Infine, vi sono piante cosiddette infestanti. Le piante infestanti sono caratterizzate da un particolare vigore, e riescono a creare danni alle altre piante fino a provocarne la morte. Queste piante devono essere messe a dimora in zone dove non possano cagionare danni alle altre. Per le piante infestanti, la distanza minima oltre la quale è bene non mettere a dimora altre piante è notevolmente più grande. [...] Relativamente al fabbisogno idrico, ogni pianta ha esigenze diverse dipendentemente dal periodo dell'anno. Una pianta, come detto in precedenza, può trovarsi nel periodo vegetativo o nel periodo di riposo. Il periodo o i periodi di fioritura e/o fruttificazione sono inclusi nel periodo vegetativo. Si suppone che una pianta richieda di essere irrigata con una certa periodicità durante il suo periodo vegetativo. Tale periodicità è espressa in giorni. Per esempio, una pianta potrebbe richiedere, in media, di essere irrigata due volte a settimana mentre un'altra ogni giorno. Durante la fase di riposo, le irrigazioni seguono un'altra periodicità. In particolare, le irrigazioni devono essere diradate (spesso fino a essere sospese) durante il periodo di riposo. Sia nel periodo vegetativo che in quello di riposo, è ovviamente importante anche la quantità d'acqua da distribuire durante le irrigazioni. Generalmente, questo dato è qualitativo, quindi si può supporre di classificare il fabbisogno idrico di ogni pianta in un intervallo di valori discreto, per esempio basso, medio, alto. [...] Quando una nuova partita di piante arriva in una sede, a ciascuna di esse viene attribuito un codice e un'anagrafica, contenente tutte le informazioni relative a nome, esigenze e caratteristiche descritte nei paragrafi precedenti. Viene inoltre associata una dimensione (derivante dall'ingombro della parte radicale) e un prezzo. Generalmente il prezzo cresce all'aumentare delle dimensioni di un esemplare. Al termine di questa fase, la pianta viene sistemata in un contenitore di dimensione adatta e collocata in un settore di una serra, su un determinato ripiano.

Frase relative al terreno

Le esigenze di terreno (detto anche substrato) sono di quattro tipi e riguardano: PH, elementi disciolti, consistenza e permeabilità. Relativamente al PH, una pianta può preferire un sottosuolo acido (piante acidofile) oppure alcalino. In generale, un determinato valore di PH è da preferire per il substrato in cui una pianta è messa a dimora, sia in vaso che in piena terra.

Frase relative agli elementi del terreno

In merito alle esigenze di elementi disciolti, una pianta può richiedere un terreno ricco di calcio, magnesio, ferro e così via. In generale, una pianta esige un substrato in cui siano presenti più micro o macro elementi, ciascuno in una determinata concentrazione. Quanto più il substrato in cui una pianta si colloca è prossimo a tali concentrazioni, tanto migliore sarà lo stato generale della pianta durante la sua coltivazione. [...] Fra le esigenze sopra descritte, quella legata agli elementi disciolti può senz'altro essere annoverata fra le più importanti. Infatti, quando si progetta di mettere a dimora una determinata pianta in prossimità di un'altra pianta (in piena terra o in grandi vasi), non si possono mettere a dimora piante che entrano in conflitto fra di loro per l'approvvigionamento degli elementi presenti nel terreno. Con il termine conflitto si intende qui una particolare condizione in cui una pianta sottrae a un'altra alcuni elementi ad essa necessari per vivere, portandola a sofferenza.

Frase relative alle componenti del terreno

Inoltre, il substrato ideale per una pianta può avere diverse consistenze. Un substrato può essere argilloso, sabbioso, laterico ecc. La consistenza, così come la permeabilità, sono determinate dai componenti del substrato quali, per esempio, sabbia, torba e ghiaia. Un substrato può quindi essere sciolto, compatto, argilloso, sabbioso, oppure un ibrido dei precedenti. In un dato substrato, i componenti sono presenti in una determinata percentuale che ne determina sia la consistenza che la permeabilità. Per esempio, terreni ricchi di argilla sono molto compatti e poco permeabili. D'altra parte, terreni sabbiosi tenderanno a rimanere più friabili e a mantenere un elevato livello di drenaggio.

Frase relative alle patologie

In determinati periodi dell'anno, una pianta è suscettibile a particolari patologie, con una certa probabilità ed entità. Le patologie delle piante si dividono in base all'agente patogeno che le causa, fra i quali vi sono insetti, acari, crittogame, funghi, virus e batteri.

Frase relative ai sintomi

Ciascuna patologia è caratterizzata da sintomi che sono evidenti semplicemente osservando la pianta. I sintomi di ogni patologia sono classificati nel database mediante un codice, una descrizione testuale e una o più immagini, in modo tale che apparati smart riescano a contribuire alla diagnosi tempestiva delle patologie mediante pattern recognition. L'insieme dei sintomi osservati su una pianta è detto sintomatologia. Esempi di sintomatologia includono l'arricciamento delle foglie, la loro caduta, il loro cambio di colore, oppure cambiamenti relativi alla turgidità della pianta nel suo complesso, così come l'interruzione della fioritura o la crescita stentata. Si noti che patologie diverse possono essere contraddistinte da alcuni sintomi comuni.

Frase relative ai prodotti

Ciascuna patologia può essere combattuta mediante lotta chimica oppure mediante lotta biologica. I principi di lotta biologica non sono qui presi in considerazione. Relativamente alla lotta chimica, esistono prodotti anticrittogamici per i funghi, insetticidi per la lotta agli insetti, acaricidi contro gli acari e così via. Ogni prodotto si basa su un principio attivo o sulla combinazione di più principi attivi, presenti in esso in determinate concentrazioni. Un prodotto chimico può essere somministrato in modi diversi per combattere una patologia. Alcuni possono essere somministrati per irrigazione nel terreno, altri per nebulizzazione, altri ancora in ambedue le modalità. In generale, lo stesso prodotto chimico può essere capace di combattere uno o più agenti patogeni. In particolare, si parla di insetticidi selettivi oppure ad ampio spettro. Un insetticida ad ampio spettro combatte più di un agente patogeno ed è quindi indicato per più patologie. Per esempio, un attacco di afidi si può combattere con Confidor, un insetticida prodotto dalla Bayer. Lo stesso Confidor combatte anche altri insetti come metcalfa, psilla, cicalina, dorifora e altri. Per ogni insetto da debellare, Confidor ha un dosaggio consigliato. Un insetticida ad ampio spettro non può essere utilizzato in alcuni periodi dell'anno. Per esempio, Confidor non può essere utilizzato durante il periodo di fioritura per non provocare la morte di intere colonie di api. Se un prodotto chimico è utilizzato su piante destinate alla produzione di frutti per l'alimentazione dell'uomo, deve essere

atteso un tempo minimo prima di poter consumare i frutti. Infatti, solo quando tale tempo è trascorso, il prodotto chimico si è dissolto ed è divenuto praticamente innocuo per la salute dell'uomo.

Frase relative alle sedi

La coltivazione delle piante all'interno dell'azienda avviene mediante principi smart. L'azienda è dislocata sul territorio nazionale in più sedi. Il personale nelle sedi è limitato e la tecnologia è utilizzata in maniera massiva in molte delle ordinarie attività. Ciascuna sede è contraddistinta da un codice, da un nome, da un indirizzo e da un numero di dipendenti.

Frase relative alle serre

Le sedi si compongono generalmente di più serre, nelle quali sono coltivate le piante. Ogni serra possiede un codice, un nome, un indirizzo, delle dimensioni (larghezza, lunghezza, altezza), un numero massimo di piante ospitabili e un numero di piante attualmente presenti. Ciascuna serra è munita di impianto di irrigazione, illuminazione, regolazione della temperatura e umidità ambientale. Queste informazioni sono salvate nel database a intervalli regolari, in modo tale da poter studiare, a posteriori, le cause degli eventi che causano danni alle piante.

Frase relative alle sezioni

Ogni serra è suddivisa in sezioni. Ogni sezione ha un codice, un nome, una capienza in termini di numero di piante ospitabili e un livello di riempimento in termini di piante effettivamente presenti. Ciascuna sezione è caratterizzata da un livello di irrigazione, illuminazione, umidità e temperatura, ed è munita di ripiani sui quali sono disposte le piante.

Frase relative ai ripiani

Ogni ripiano possiede un codice e un insieme di contenitori. I ripiani condividono l'illuminazione, l'umidità e la temperatura della sezione alla quale appartengono, ma non il livello di irrigazione.

Frase relative ai contenitori

Ogni pianta ospitata nella serra è collocata in un contenitore. [...] Ogni contenitore ha un codice, una dimensione in termini di superficie occupata, un livello di idratazione e una particolare composizione del substrato. Queste informazioni sono rilevate dai sensori del contenitore. Il livello di irrigazione, che determina il livello di idratazione, è diverso da un contenitore all'altro ed è regolato automaticamente in base ai dati provenienti dai sensori presenti in ogni contenitore.

Frase relative alla salute di una pianta

Ogni pianta è insomma costantemente monitorata mediante tali sensori che sono posizionati nel terreno e che rilevano in tempo reale informazioni relative a umidità, concentrazione dei principali elementi (potassio, magnesio ecc.) e anche formazioni riconducibili a marcescenza o a sofferenza radicale. In particolare, queste formazioni possono essere rilevate mediante esalazioni gassose di elementi putrescenti, come il metano, oppure di altre sostanze che si sviluppano nel terreno, ad esempio elevati tassi di ammoniaca. Giornalmente, i dati relativi a ogni contenitore sono misurati e le informazioni sono memorizzate nel database, senza rimuovere i dati relativi alle misurazioni precedenti.

Frase relative ai report di diagnostica

Per la lotta alle patologie, in ogni serra è in funzione un sistema di prevenzione e cura delle principali malattie delle piante che vi risiedono in attesa di essere vendute. L'identificazione delle patologie delle piante avviene mediante microcamere collocate su droni che monitorano costantemente gli esemplari della serra. Come detto precedentemente, ogni pianta è monitorata anche grazie ai sensori posizionati nel terreno. Ogniqualvolta il sistema rileva una patologia in vivaio, a fine giornata, compila un report di diagnostica nel quale segnala gli esemplari colpiti e propone una serie di patologie possibili, ordinate per attinenza con i sintomi rilevati. Inoltre, il report di diagnostica contiene informazioni relative all'istante in cui è stata rilevata una potenziale patologia. In particolare, nel report di diagnostica è presente il codice della pianta colpita, le informazioni circa le principali sostanze rilevate nel terreno e in quali concentrazioni, e l'insieme di sintomi rilevati.

Frase relative ai trattamenti

La diagnosi finale è fatta da un esperto che decide anche il trattamento da mettere in pratica. In particolare, tutte le piante colpite dalla stessa patologia vengono solitamente isolate e messe in quarantena in apposite sezioni della serra. Una volta isolate, irrigatori a pioggia erogano un prodotto chimico in una determinata dose, effettuando quindi un trattamento. I trattamenti subiti da ogni pianta sono memorizzati nel database mediante un codice trattamento, il codice della pianta coinvolta, il prodotto somministrato, in quale dose e in quale data. Le piante vengono successivamente monitorate e, in caso di inefficacia del primo trattamento, sottoposte a trattamenti aggiuntivi finché la patologia non viene debellata. Solo allora la pianta può essere collocata nuovamente nel contenitore originario che, nel frattempo, non deve essere rimpiazzato da altre piante per evitare potenziali contaminazioni.

Frase relative agli accounts

Ogni cliente ha un account al quale collega le sue informazioni di anagrafica, quali nome, cognome, un nickname, una password, un indirizzo di posta elettronica, una domanda segreta per il recupero della password (con relativa risposta), e la città di residenza. Ad ogni account è anche collegato un profilo nel quale il cliente esprime quali sono le piante delle quali è più appassionato.

Frase relative alle schede

Inoltre, il profilo è connesso a schede, ciascuna relativa a una pianta (esemplare) che il cliente ha acquistato dall'azienda. Ciascuna scheda è contraddistinta da un codice, il nome della pianta a cui si riferisce, la dimensione della stessa al momento dell'acquisto, la data d'acquisto e l'attuale collocazione: piena terra o vaso. Se è stata collocata in un vaso, devono essere presenti le dimensioni dello stesso.

Frase relative ai posts nel forum

L'area social contiene anche un forum di discussione nel quale gli utenti possono inserire post per chiedere consigli circa l'acquisto di nuove piante, per la cura di patologie che coinvolgono esemplari acquistati, o per condividere le loro esperienze e piccoli successi nel giardinaggio. I post sono pubblicati dagli utenti in diversi thread del forum. Ciascun post è contraddistinto dal nickname dell'utente che lo ha pubblicato, da un testo, dal timestamp di pubblicazione, dal thread del forum in cui è pubblicato e da eventuali link a contenuti multimediali (URL a contenuti ospitati su server di terze parti). I post ricevono risposte da altri utenti del forum. I post di risposta possono essere valutati dagli altri utenti con un giudizio (da una a cinque stelle). Un utente acquisisce una determinata credibilità all'interno del forum in base al numero di post pubblicati nel tempo e ai giudizi ottenuti dagli altri utenti che valutano i suoi post. La politica di attribuzione della credibilità (cioè, di un punteggio) e la sua evoluzione temporale è lasciata a discrezione dello studente.

Frase relative agli ordini

Una volta selezionata la pianta da acquistare, il sistema crea un nuovo ordine. Un ordine è caratterizzato da un codice ordine, un codice cliente (lo si può pensare coincidente con il nickname del profilo), un codice pianta acquistata, un istante temporale di effettuazione dell'ordine e uno stato. Lo stato di un ordine passa attraverso i seguenti valori: in processazione, in preparazione, spedito, evaso. Un ordine deve necessariamente seguire la precedente sequenza di stati. Quando un ordine passa nello stato evaso, viene creata automaticamente una scheda all'interno del suo profilo. Se un esemplare non è attualmente presente in alcuna delle sedi, l'utente ha ugualmente la possibilità di completare un acquisto. L'ordine relativo a tale acquisto è memorizzato, in stato pendente. Sostanzialmente, ordini di questo tipo non sottraggono credito al cliente, ma sono utili all'azienda per capire quali prodotti (piante) devono essere a breve reperite dai fornitori per poter far avanzare lo stato degli ordini pendenti. Non appena gli articoli relativi a ordini pendenti risultano di nuovo disponibili un ordine pendente passa in stato in processazione, e da lì negli altri stati, secondo l'ordine sopra descritto.

Frase relative alle manutenzioni

Esempi di manutenzione includono la potatura, il rinvaso e la concimazione. [...] Durante il periodo vegetativo, le piante necessitano generalmente anche di interventi di concimazione. Un intervento di concimazione si basa sulla somministrazione di uno o più macro (o micro) elementi da disciogliere nel terreno o da nebulizzare sulla parte aerea della pianta. Gli interventi di concimazione che una pianta richiede possono essere molteplici durante il periodo vegetativo. Ciascun intervento di concimazione è caratterizzato da un periodo dell'anno in cui è opportuno effettuarlo, dalle sostanze richieste, dalla rispettiva quantità (peso) e da una periodicità. Ad esempio, un intervento nel quale si somministra del chelato di ferro potrebbe essere richiesto due volte a distanza di tre mesi per garantire un'ottimale

disponibilità di ferro nel terreno durante tutta la fase vegetativa. Non è detto che in ambedue le somministrazioni la quantità da somministrare sia la stessa. [...] Una pianta, dipendentemente dal suo genere di appartenenza, è caratterizzata da un indice di accrescimento in base al quale essa può più o meno espandere la sua parte aerea e la sua parte radicale nel tempo. Ciò determina l'esigenza di interventi diversificati, da pianificare nel tempo, per gestire la salute e il corretto sviluppo delle piante, soprattutto per adattarne le dimensioni e renderle consone al luogo in cui sono messe a dimora. In particolare, una pianta può necessitare di uno o più interventi di potatura durante l'anno. Vi sono diverse tipologie di potatura. Le potature possono infatti essere destinate al contenimento delle dimensioni, all'aumento di produzione di fiori e frutti, alla rimozione di parti morte o logore, oppure al rinnovo della pianta, come per esempio la capitozzatura, un intervento drastico di potatura che si effettua ad alberi di dimensioni ragguardevoli per ridefinirne la chioma, tagliando le principali diramazioni del tronco, in prossimità dello stesso. Questo tipo di potatura è spesso visibile sui viali alberati. Un intervento di potatura è caratterizzato, quindi, da una tipologia e da uno o più periodi dell'anno in cui si può effettuare. La stessa tipologia di potatura può non essere effettuabile nello stesso periodo dell'anno su piante di genere diverso. Caratteristiche analoghe valgono per i rinvasi, che consistono nella sostituzione del vaso che contiene una pianta con uno di capienza maggiore. Anche i rinvasi possono essere effettuati in determinati periodi dell'anno in determinate piante. Altre, invece, tollerano rinvasi durante tutto il periodo dell'anno. [...] L'area manutenzione si occupa di gestire gli interventi di manutenzione su richiesta, programmata e automatica. La manutenzione su richiesta è segnalata mediante il sito web, tramite un'apposita richiesta effettuata dall'utente. La richiesta è collegata a una delle schede presenti nel profilo dell'utente, dalla quale è possibile capire per quale pianta il cliente richiede manutenzione. Ad ogni richiesta il cliente associa una scadenza temporale entro la quale desidera che l'intervento di manutenzione sia portato a termine. Infine, per ogni richiesta, il cliente esprime di che tipo di manutenzione necessita. [...] A seguito dell'inserimento di una richiesta di manutenzione del database, la richiesta viene processata (possibilmente entro i tempi stabiliti dal cliente). [...] Nella manutenzione programmata, l'utente stabilisce quali delle sue piante desidera siano oggetto di manutenzione programmata nel corso del tempo. Per esempio, se l'utente possiede alcuni alberi da frutto, può desiderare che ogni anno, alla fine dell'inverno, i suoi alberi da frutto siano potati. In questo modo, l'utente non dovrà più preoccuparsi di avvertire l'azienda, la quale provvederà autonomamente a eseguire i regolari interventi di potatura. [...] La manutenzione automatica si basa sull'indice di accrescimento delle piante, sui periodi di fioritura e sulle loro esigenze in genere. In particolare, in base alle piante possedute da un utente, ciascuna riconducibile a una scheda nel relativo profilo, il sistema visualizza delle notifiche sul social, nei vari profili, suggerendo interventi di manutenzione ai clienti. Le notifiche sono visualizzate, per gli utenti che lo desiderano, nelle schede che essi indicano come oggetto di manutenzione automatica. Per esempio, le siepi e le bordure si potano circa due volte all'anno. Quindi, un cliente che possiede una siepe in alloro, avrà degli interventi automatici che gli saranno notificati, se ne ha espressa la volontà, ogni anno in Settembre e in Marzo. Inoltre, in base al tipo di pianta posseduta e alle sue vulnerabilità, saranno suggeriti anche interventi di concimazione, nonché trattamenti chimici nei periodi con maggiore probabilità di attacco di parassiti, crittogame e così via. Se un cliente non risponde alla notifica prenotando un intervento di manutenzione, o comunque non afferma di avervi provveduto autonomamente, il sistema continua a fare le sue stime, usando fra le altre cose, gli indici di accrescimento e manutenzione, tenendo quindi traccia dell'entità dell'intervento di manutenzione che diviene tanto più rilevante quanto più tempo si lascia passare dalla prima notifica ricevuta. Gli interventi di manutenzione da effettuare (e già effettuati) sono memorizzati nel database e sono sempre visibili nel profilo del cliente. Il cliente può, inoltre, inserire nel suo profilo interventi di manutenzione da egli/ella effettuati senza coinvolgere l'azienda. In particolare, il cliente può aver provveduto autonomamente a rinvasare una certa pianta, a concimarne un'altra e a potarne un'altra ancora. Il profilo contiene, quindi, lo stato aggiornato di tutte le sue piante, mediante le schede. Ogni intervento di manutenzione ha un costo che è inserito nella scheda dopo che il personale ha valutato la situazione con un sopralluogo.

Frase relative ai settori

Inoltre, ogni spazio può essere a sua volta diviso in settori. Ogni settore destinato alla presenza di piante può consistere di piena terra, oppure può essere pavimentato, ma contenere vasi. Anche un settore composto di piena terra può contenere vasi. I vasi sono caratterizzati da una dimensione e da un materiale. I settori possono essere pensati come poligoni, così come il giardino in sé. Ogni settore ha una esposizione (cioè si rivolge verso un punto cardinale) ed è più o meno esposto alla luce diretta del Sole. Non è detto che settori esposti a sud siano sempre esposti al Sole perché potrebbero essere presenti verande, pergolati, e così via. L'utente, oltre a specificare il punto cardinale di ogni settore, deve quindi indicare anche il numero di ore approssimativo nelle quali il settore riceve la luce diretta del Sole. Una volta costruito il modello del giardino, il sistema assegna un codice a ogni settore e a ogni vaso.

Frase relative agli spazi

L'area progettazione è raggiungibile dall'area riservata (profilo) di ciascun cliente e si occupa di memorizzare tutte le informazioni che i clienti forniscono circa i settori da progettare (quindi prive di piante) dei loro spazi verdi. In base a queste informazioni, il sistema fornisce suggerimenti su quali piante possono effettivamente essere messe a dimora nei vari settori del giardino del cliente sia in base alle necessità di queste che dal punto di vista economico. Nell'area progettazione, il sistema crea una griglia di pixel come sistema di riferimento. L'utente può tracciare un prospetto del giardino nel quale può contrassegnare spazi a verde e spazi non destinati alla presenza di piante.

Frase relative all'arredamento

Mediante funzionalità web, selezionato un settore, il cliente fa drag and drop di elementi (piante) che appaiono in un menu laterale, in modo tale da configurare il proprio giardino. [...] Completata la progettazione di un settore, deve essere possibile salvare nel database una configurazione spaziale dell'arredo del giardino, con tutte le posizioni delle varie piante collocate dal cliente. In questo modo, un login successivo può ricostruire il prospetto e mostrarlo a video. Il prospetto alla fine può essere oggetto di richiesta di preventivo all'azienda. [...] La disposizione delle piante proposta è memorizzata nel database utilizzando il sistema di riferimento in pixel creato dal sistema. [...] Il cliente può generare più versioni alternative dell'arredamento e, in seguito scegliere quale acquistare in base ai preventivi ricevuti.

2. Progettazione concettuale

2.1 Strategia progettuale

Per lo sviluppo dello schema concettuale è stata adottata una strategia di tipo *inside-out*, in quanto sono stati rappresentati prima i concetti in relazione con i concetti iniziali, per poi muoversi verso quelli più lontani attraverso una "navigazione" tra le specifiche.

Questa strategia ha il vantaggio di non richiedere passi di integrazione, ma è comunque stata necessaria più di una lettura alle specifiche.

2.2 Tabella delle entità

Entità	Descrizione	Attributi	Identificatore
Pianta	Mantiene le informazioni sia sulle piante vendute, che su quelle ancora in magazzino. Una pianta in magazzino è presente in un contenitore, una pianta venduta è presente in un vaso o è piantata in piena terra.	CodPianta, Nome, Genere, Cultivar, Infestante, DimMax, IndiceAccrescimento, DimAttuale, Prezzo, PosLuce, TipoLuce, OreLuceVegetativo, OreLuceRiposo, TempMin, TempMax, ConsistenzaTerreno, DistanzaMinConflitto, IndiceManutenzione, Fogliame, Dioica, PeriodicitIrrigRiposo, PeriodicitIrrigVegetativo, QuantalIrrigRiposo, QuantalIrrigVegetativo, NumIrrigGiornarliereRiposo, NumIrrigGiornarliereVegetativo	CodPianta
Terreno	Mantiene le informazioni su tutti i tipi di terreni, con i relativi elementi e componenti.	CodTerreno, PH, Consistenza, Permeabilita	CodTerreno
Componente	Contiene tutti i componenti possibili dei terreni, i quali determineranno la consistenza di questi ultimi.	Nome	Nome

Elemento	Contiene tutti gli elementi necessari alle piante e quelli presenti nei terreni.	Nome, PerMinConflitto	Nome
Patologia	Mantiene le informazioni sulle patologie esistenti, tra cui il periodo dell'anno in cui si manifesta, i prodotti che la combatton, gli agenti patogeni che la causano e un insieme di sintomi che essa manifesta.	CodPatologia, Entita, DataInizio, DataFine, Probabilita	CodPatologia
Sintomo	Mantiene le informazioni su tutti i sintomi esistenti, accompagnate da immagini dimostrative.	CodSintomo, Descrizione	CodSintomo
Prodotto	Mantiene le informazioni su tutti i prodotti utilizzati nei trattamenti delle piante, indicandone i principi attivi, gli agente patogeni che combatte e i periodi in cui ne è possibile l'utilizzo.	Nome, Marca, ModalitaSomm, TempoMinConsumazFrutti	Nome
Principio Attivo	Contiene tutti i principi attivi esistenti utilizzati nei prodotti.	Nome	Nome
Agente Patogeno	Contiene tutti gli agenti patogeni che causano patologie e che vengono combattuti dai prodotti.	Nome, Tipo	Nome
Sede	Mantiene le informazioni sulle sedi in cui l'azienda è dislocata sul territorio nazionale.	CodSede, Capienza, Nome, NumDipendenti, Indirizzo	CodSede
Serra	Mantiene le informazioni sulle serre contenute nelle sedi dell'azienda.	CodSerra, Nome, Indirizzo, Larghezza, Capienza, Altezza	CodSerra
Sezione	Mantiene le informazioni sulle sezioni che suddividono le serre.	CodSezione, Nome, Capienza, Irrigazione, Illuminazione, Umidita, Temperatura, Quarantena	CodSezione
Ripiano	Mantiene le informazioni sui ripiani sui quali vengono disposti i contenitori.	CodRipiano, Capienza, Irrigazione	CodRipiano
Contenitore	Mantiene le informazioni sui contenitori i quali possono contenere al più una pianta.	CodContenitore, Dimensione, Idratazione, Irrigazione	CodContenitore
Salute	Mantiene lo storico delle misurazioni relative allo stato di salute di ogni contenitore.	Timestamp, Umidita, LivelloGas, TassoAmmoniaca	Timestamp, CodContenitore (esterno)
Report Diagnostica	Segnala le piante colpite da patologie includendo le informazioni relative all'istante in cui è stata rilevata e i sintomi rilevati.	Timestamp	Timestamp, CodPianta (esterno)
Vaso	Mantiene le informazioni sui vasi che contengono piante acquistate.	CodVaso, Materiale, DimensioneX, DimensioneY	CodVaso
Account	Mantiene le informazioni sui clienti, come le piante acquistate, le piante per cui si ha una	Nickname, Nome, Cognome, Password, Email, DomandaSegreta, RispostaSegreta, CittaResidenza,	Nickname

	passione, gli ordini, le attività sul forum e quelle di garden design.	NumGiudizi, NumPostPubblicati, GiudizioMedio, Credibilita	
Scheda	Mantiene le informazioni relative all'acquisto e alla collocazione spaziale delle piante acquistate dai clienti.	CodScheda, NomePianta, DataAcquisto, Settore, Collocazione, DimensioneAllAcquisto, PosX, PosY	CodScheda
Post	Mantiene le informazioni sui post pubblicati dai clienti sul forum.	Timestamp, Giudizio, Testo	Timestamp, Nickname (esterno)
Thread	Mantiene le informazioni sui thread che contengono i vari post degli utenti.	CodThread, Nome	CodThread
Ordine	Mantiene le informazioni sugli ordini delle piante effettuati dagli utenti.	CodOrdine, Timestamp, Stato	CodOrdine
Report Ordini	Segnala all'azienda quali piante è conveniente o meno ordinare dai fornitori, includendo eventuali informazioni sul clima interessato in caso si trattino di piante da non ordinare.	CodRepOrdini, DaOrdinare, Clima	CodRepOrdini
Report Assunzioni	Segnala il numero di dipendenti da assumere in specifici periodi.	CodReportAss, CodSede, NumDipendenti, Indeterminato	CodReportAss
Manutenzione	Mantiene le informazioni sulle manutenzioni (Concimazione, Potatura, Rinvaso e Trattamento) effettuate, non effettuate e richieste dalle piante.	CodManutenzione, Costo, Tipo, Scadenza, NumIntervAnnuali, Svolta, DataInizio	CodManutenzione
Report Manutenzione	Segnala periodicamente, per ogni specie di pianta, il costo totale fino a tal periodo per ogni tipo di manutenzione effettuata.	CostoTot, Tipo, NumSnapshot	Tipo, NumSnapshot, CodPianta (esterno)
Spazio	Mantiene le informazioni sul giardino che il cliente ha a disposizione per la disposizione delle piante.	CodSpazio, Tipo	CodSpazio
Settore	Mantiene le informazioni relative ai settori di uno spazio, che contengono le piante.	CodSettore, Tipo, Esposizione, NumOreLuce	CodSettore
Vertice	Entità di supporto dedicata alla memorizzazione delle forme geometriche dei vari settori.	PosX, PosY	PosX, PosY
Arredamento	Mantiene le informazioni riguardo la predisposizione delle piante (in vaso o in terra piena) di uno spazio verde. È possibile memorizzare più versioni.	Versione	Versione, CodSpazio (esterno)

2.3 Tabella delle relazioni

Relazione	Descrizione	Entità coinvolte	Attributi
PianteTerreno	Associa ad ogni pianta il terreno nella quale è piantata.	Pianta (1, 1) Terreno (0, N)	
ElementiRerreno	Associa ad ogni terreno gli elementi in esso contenuti, indicandone la concentrazione in mg/m^3 . Determinano il PH del terreno.	Terreno (1, N) Elemento (0, N)	Concentrazione
ComponentiTerreno	Associa ad ogni terreno i componenti in esso contenuti, indicandone la concentrazione in mg/m^3 . Determinano la consistenza e la permeabilità del terreno.	Terreno (1, N) Componente (0, N)	Concentrazione
TerrenoContenitore	Associa ad ogni contenitore, il terreno in esso contenuto.	Terreno (0, N) Contenitore (1, 1)	
ContenitorePianta	Associa ad ogni pianta in magazzino (non venduta) il contenitore in cui essa è collocata.	Pianta (0, 1) Contenitore (0, 1)	
ElementiNecessariPianta	Associa ad ogni pianta tutti quegli elementi di cui ha bisogno per vivere, indicandone la concentrazione in mg/m^3 .	Pianta (1, N) Elemento (0, N)	Concentrazione
DatiSaluteContenitore	Associa ad ogni contenitore uno storico di dati relativi alla salute della pianta contenuta.	Salute (1, 1) Contenitore (0, N)	
ElementiStatoSalute	Associa ad ogni dato relativo alla salute di una pianta, gli elementi presenti nel terreno nell'istante temporale in cui sono state effettuate le misurazioni.	Salute (1, N) Elemento (0, N)	Concentrazione
ContenitoriRipiano	Associa ad ogni ripiano i contenitori che esso contiene.	Contenitore (1, 1) Ripiano (0, N)	
RipianiSezione	Associa ad ogni sezione i ripiano che essa contiene.	Ripiano (1, 1) Sezione (0, N)	
SezioniSerra	Associa ad ogni serra le sezioni che essa contiene.	Sesione (1, 1) Serra (1, N)	
SerreSede	Associa ad ogni sede le serre che essa contiene.	Serra (1, 1) Sede (1, N)	
SintomiPatologia	Associa ad ogni patologia uno o più sintomi che questa produce.	Patologia (1, N) Sintomo (0, N)	
ImmaginiSintomi	Associa ad ogni sintomo una o più immagini dimostrative.	Sintomo (1, N) Immagine (0, N)	
ProdottiPatologia	Associa ad ogni patologia uno o più prodotti necessari ad estinguerla.	Patologia (1, N) Prodotto (0, N)	
AgentiPatogeniPatologia	Associa ad ogni patologia uno o più agenti patogeni che ne rappresentano la causa.	Patologia (1, N) AgentePatogeno (0, N)	

PrincipiAttiviProdotto	Associa ad ogni prodotto uno o più principi attivi che lo compongono, indicandone la concentrazione in <i>mg/l</i> .	Prodotto (1, N) PrincipioAttivo (1, N)	Concentrazione
ProdottoCombatte	Associa ad ogni prodotto uno o più agenti patogeni combattuti da quest'ultimo.	Prodotto (1, N) AgentePatogeno (0, N)	Dosaggio
PeriodiProdotto	Associa ad ogni prodotto uno o più periodi durante i quali è possibile farne uso.	Periodo (0, N) Prodotto (1, N)	
ReportDiagnosticaPianta	Associa ad ogni report di diagnostica la pianta a cui fa riferimento.	Pianta (0, N) ReportDiagnostica (1, 1)	
TerrenoReport	Associa ad ogni report di diagnostica il terreno in cui la pianta è presente.	Terreno (0, N) ReportDiagnostica (1, 1)	
SintomiReport	Associa ad ogni report di diagnostica uno o più sintomi dimostrati dalla pianta.	ReportDiagnostica (1, N) Sintomo (0, N)	
PatologieReport	Associa ad ogni report di diagnostica una o più patologie possibili (la diagnosi).	ReportDiagnostica (1, N) Patologia (0, N)	
PeriodiPianta	Associa ad ogni pianta i suoi periodi, indicandone il tipo (di fioritura, di fruttificazione o di riposo).	Pianta (1, N) Periodo (0, N)	Tipo
PeriodiManutenzione	Associa ad ogni manutenzione uno o più periodi durante i quali questa è consentita.	Periodo (0, N) Manutenzione (1, N)	
ManutenzioniPianta	Associa ad ogni manutenzione (se svolta) la pianta a cui è stata effettuata.	Pianta (0, N) Manutenzione (0, 1)	
ProdottiTrattamento	Associa ad ogni trattamento uno o più prodotti utilizzati, indicandone la concentrazione in <i>mg/l</i> .	Prodotto (0, N) Trattamento (1, N)	Dose
EsigenzaConcimazionePianta	Associa ad ogni pianta nessun o più concimazioni necessarie per il suo mantenimento in vita.	Pianta (0, N) Concimazione (0, N)	
ReportManutenzionePianta	Associa ad ogni report di manutenzione la pianta a cui esso fa riferimento.	Pianta (0, N) ReportManutenzione (1, 1)	
PeriodoReportAssunzioni	Associa ad ogni report assunzioni uno o più periodi nel caso il report indichi assunzioni non a tempo indeterminato.	Periodo (0, N) ReportAssunzioni (0, 1)	
PianteAppassionatoAccount	Associa ad ogni account zero o più piante alle quali il cliente indica di essere appassionato.	Pianta (0, N) Account (0, N)	
PianteOrdine	Associa ad ogni ordine una o più piante al quale esso fa riferimento.	Ordine (1, N) Pianta (0, 1)	
OrdiniCliente	Associa ad ogni ordine il cliente che lo ha generato.	Ordine (1, 1) Account (0, N)	

PianteReportOrdini	Associa ad ogni report degli ordini le piante a cui esso fa riferimento.	Pianta (0, N) ReportOrdini (1, N)	Quantita
PostsAccount	Associa ad ogni post l'account che lo ha creato.	Account (0, N) Post (1, 1)	
PostsThread	Associa ad ogni post il thread di appartenenza.	Post (1, 1) Thread (0, N)	
MediaPost	Associa ad ogni post zero o più contenuti multimediali allegati.	Post (0, N) Media (1, N)	
SchedeAccount	Associa ad ogni account zero o più schede relative alle piante acquistate dal cliente.	Account (0, N) Scheda (1, 1)	
SchedaPianta	Associa ad ogni pianta venduta la scheda del profilo ad essa associata.	Pianta (0, 1) Scheda (1, 1)	
SpaziUtente	Associa ad ogni spazio verde l'utente che lo possiede.	Account (0, N) Spazio (1, 1)	
SettoriSpazio	Associa ad ogni spazio verde zero o più settori.	Spazio (0, N) Settore (1, 1)	
VerticiSettore	Assegna ad ogni settore almeno tre vertici per memorizzarne la forma.	Settore (3, N) Vertice (0, N)	
ArredamentiSpazio	Associa ad ogni arredamento lo spazio a cui fa riferimento. Ogni spazio può possedere più versioni di arredamenti.	Spazio (0, N) Arredamento (1, 1)	
PianteArredamentoInPienaTerra	Associa ad ogni arredamento zero o più piante in piena terra, indicandone la posizione.	Pianta (0, N) Arredamento (0, N)	PosX, PosY
VasiArredamento	Associa ad ogni arredamento zero o più vasi, indicandone la posizione e la eventuale pianta contenuta.	Vaso (0, N) Arredamento (0, N) Pianta (0, N)	PosX, PosY
VasoScheda	Associa ad ogni scheda l'eventuale vaso in cui la pianta è realmente collocata.	Scheda (0, 1) Vaso (0, 1)	
SomministrazioniConcimazione	Associa ad ogni concimazione più di un elemento da utilizzare come concime, indicandone la quantità per ogni iterazione.	Concimazione (1, N) Elemento (0, N)	Iterazione, Quantita

2.4 Schema concettuale

Lo schema concettuale risulta così formato:

3. Ristrutturazione del diagramma E-R

3.1 Analisi delle ridondanze

Di seguito le ridondanze ritrovate dopo l'analisi dello schema concettuale, raggruppate per tipologia. Il loro mantenimento o meno verrà discusso nel paragrafo [5.3 "Considerazione sulle ridondanze già presenti"](#).

3.1.1 Attributi derivabili da altri attributi della stessa entità

- **Ridondanza 1**

L'attributo *Credibilita* dell'entità *Account* può essere calcolata tramite gli attributi *GiudizioMedio*, *NumPostPubblicati* e *NumGiudizi* tramite la seguente relazione:

$$Credibilita = GiudizioMedio * NumGiudizi - (NumPostPubblicati - NumGiudizi)$$

ovvero

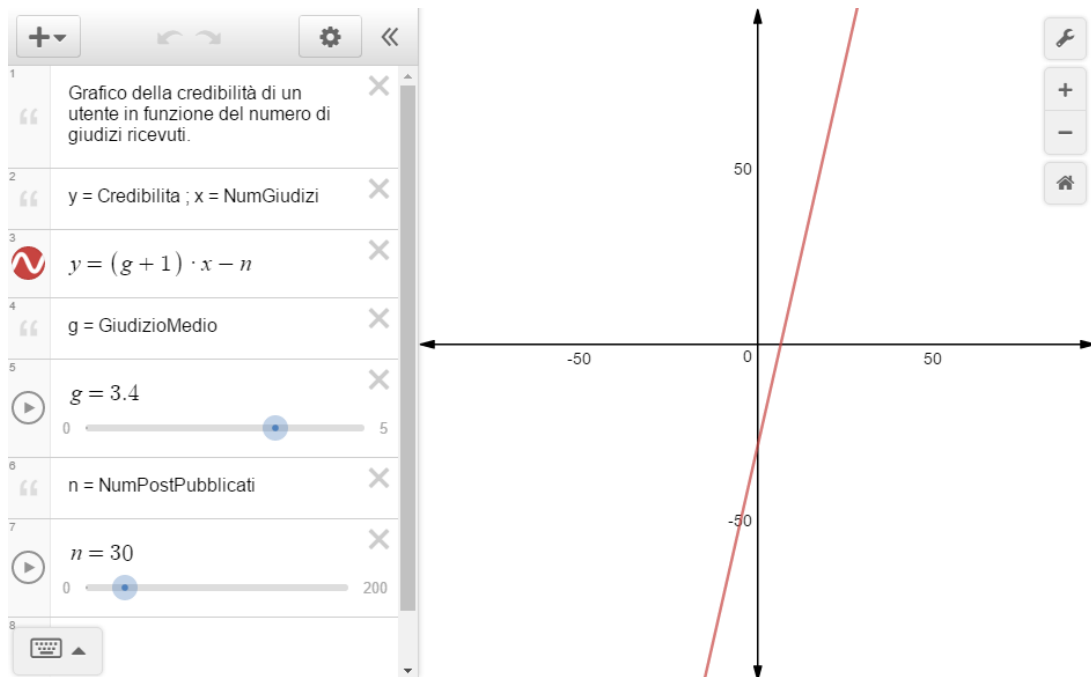
$$Credibilita = (GiudizioMedio + 1) * NumGiudizi - NumPostPubblicati$$

La seguente relazione è stata ottenuta tenendo conto di due fattori:

1. La credibilità di un utente dev'essere direttamente proporzionale al giudizio medio di quest'ultimo e al numero di giudizi ricevuti.
A parità di giudizio medio, è più credibile un utente che ha ricevuto un maggior numero di giudizi. Questo è rappresentato dal minuendo della prima espressione.
2. Più è alta la forbice tra il numero dei post pubblicati e il numero di giudizi ricevuti, più l'utente è stato "ignorato" dal resto della community. Questo deve influire negativamente sulla credibilità. Questo è rappresentato dal sottraendo della prima espressione.

È possibile perciò rimuovere quest'attributo senza incorrere in una perdita di informazione.

È possibile interagire con i parametri di questa relazione su desmos.com/calculator/4e4te5qatt, che mostra il grafico della credibilità in funzione del numero di giudizi ricevuti.



3.1.2 Attributi derivabili da attributi di un'altra entità

- **Ridondanza 2**

L'attributo *NumGiudizi* dell'entità *Account* è deducibile contando i valori diversi da NULL dell'attributo *Giudizio* delle entità *Post* collegate tramite la relazione *PostsAccount*.

- **Ridondanza 3**

L'attributo *GiudizioMedio* dell'entità *Account* è deducibile effettuando la media tra gli attributi *Giudizio* delle entità *Post* collegate tramite la relazione *PostsAccount*.

- **Ridondanza 4**

L'attributo *NomePianta* dell'entità *Scheda* è deducibile dall'attributo *Nome* dell'entità *Pianta* ad essa collegata.

3.1.3 Attributi derivabili dal conteggio delle occorrenze

- **Ridondanza 5**

L'attributo *NumPostPubblicati* dell'entità *Account* è deducibile contando le partecipazioni alla relazione *PostsAccount*.

- **Ridondanza 6**

L'attributo *Svolta* dell'entità *Manutenzione* è deducibile controllando la partecipazione di quest'ultima nell'associazione *ManutenzioniPianta*, che associa ad ogni manutenzione svolta la pianta ad essa relativa.

3.1.4 Associazioni derivabili dalla composizione di altre associazioni

- **Ridondanza 7**

L'associazione *TerrenoContenitore* risulta ridondante nel caso in cui non interessi che terreno è presente nei contenitori vuoti. Questo può essere legittimo in quanto, in caso di inserimento di una nuova pianta in magazzino, se il terreno attualmente presente nel contenitore non è compatibile con la pianta da collocare, esso può essere svuotato o più semplicemente può essere scelto un altro contenitore.

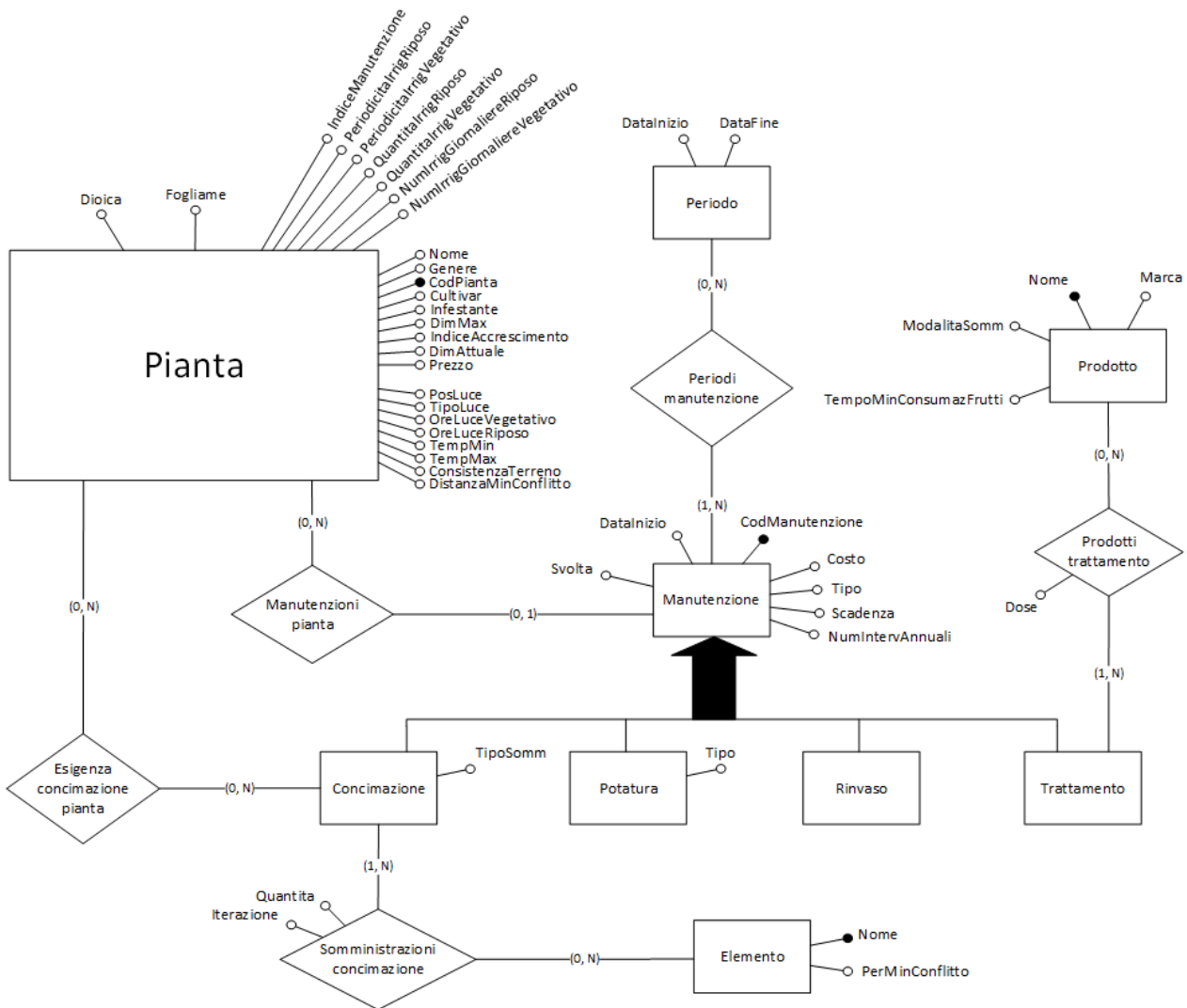
3.2 Traduzione degli attributi multivalore

Gli attributi multivalore sono stati fin da subito rappresentati per mezzo di nuove entità connesse all'entità di partenza. Alcune delle entità di supporto create apposta per evitare l'utilizzo di attributi multivalore sono:

- Immagine
- Periodo
- Media

3.3 Traduzione delle generalizzazioni

L'unica generalizzazione presente è totale ed esclusiva, che vede come entità padre l'entità *Manutenzione*.



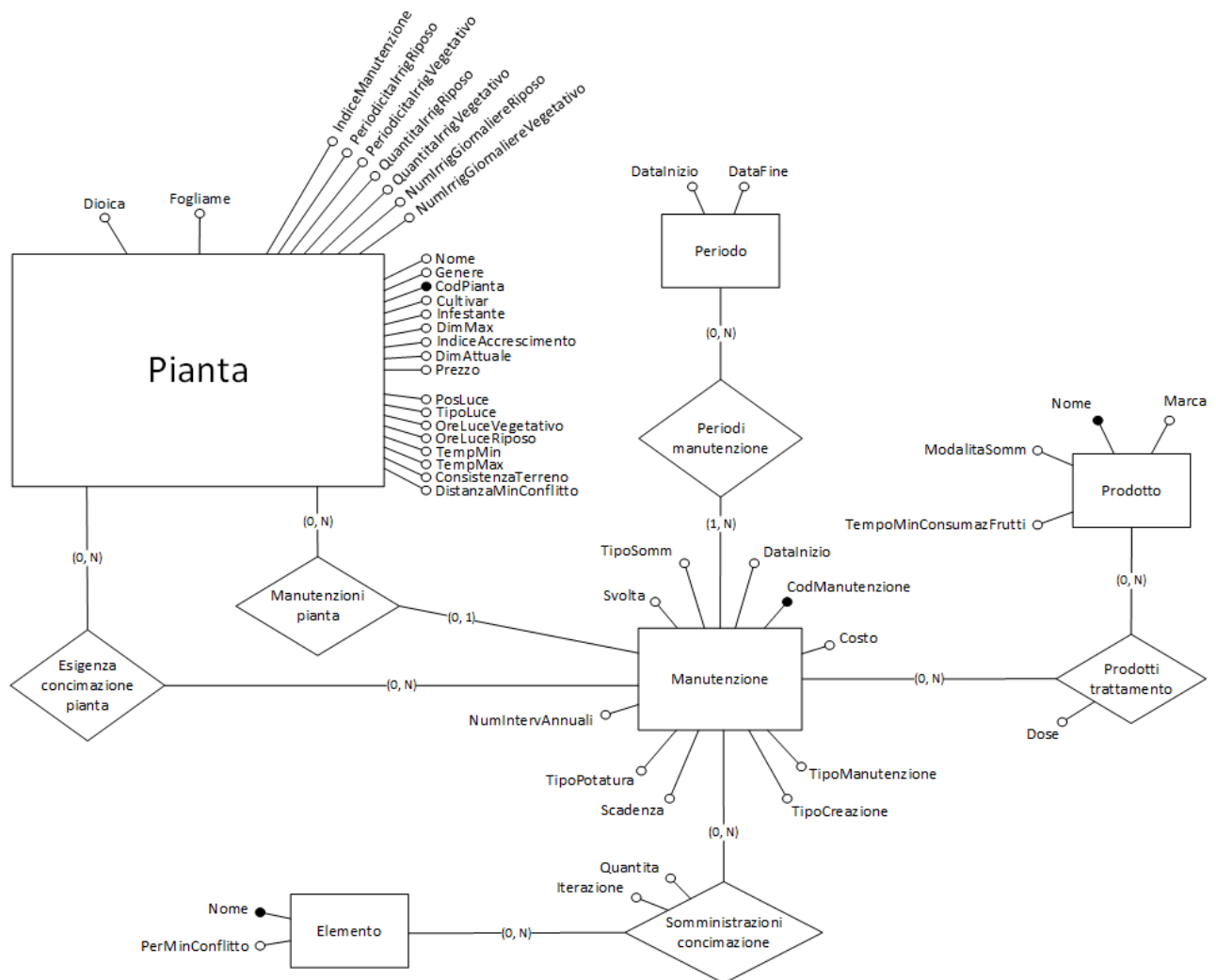
Si opta per l'accorpamento dei figli sul padre, in quanto le 4 entità figlie hanno molti attributi a comune. L'attributo *TipoSomm* dell'entità *Concimazione* verrà ereditata dall'entità padre, ed essa avrà significato solo nel caso in cui l'entità *Manutenzione* rappresenti un intervento di concimazione; avrà valore *NULL* altrimenti. L'attributo *Tipo* dell'entità *Potatura* verrà ereditata dall'entità padre e ne verrà cambiato il nome in *TipoPotatura* per evitare ambiguità con l'attributo *Tipo* dell'entità *Manutenzione* che specifica se l'intervento di manutenzione è su richiesta, programmata o automatica (questo attributo verrà rinominato in *TipoCreazione*). L'associazione *Esigenza concimazione pianta* collegata all'entità *Concimazione* verrà

ereditata dall'entità padre e non ha bisogno di ulteriori modifiche.

Le rimanenti due associazioni, *Prodotto trattamento* e *Somministrazioni concimazione*, invece, per essere adottate dall'entità padre avranno bisogno di un cambio di cardinalità: potendo l'entità *Manutenzione* rappresentare altri tipi di intervento oltre al *Trattamento*, potrebbe non avere associato nemmeno un *Prodotto*, e, per lo stesso motivo, l'entità *Manutenzione* potrebbe non avere associato nemmeno un *Elemento*. La cardinalità in questione diventeranno quindi (0, N).

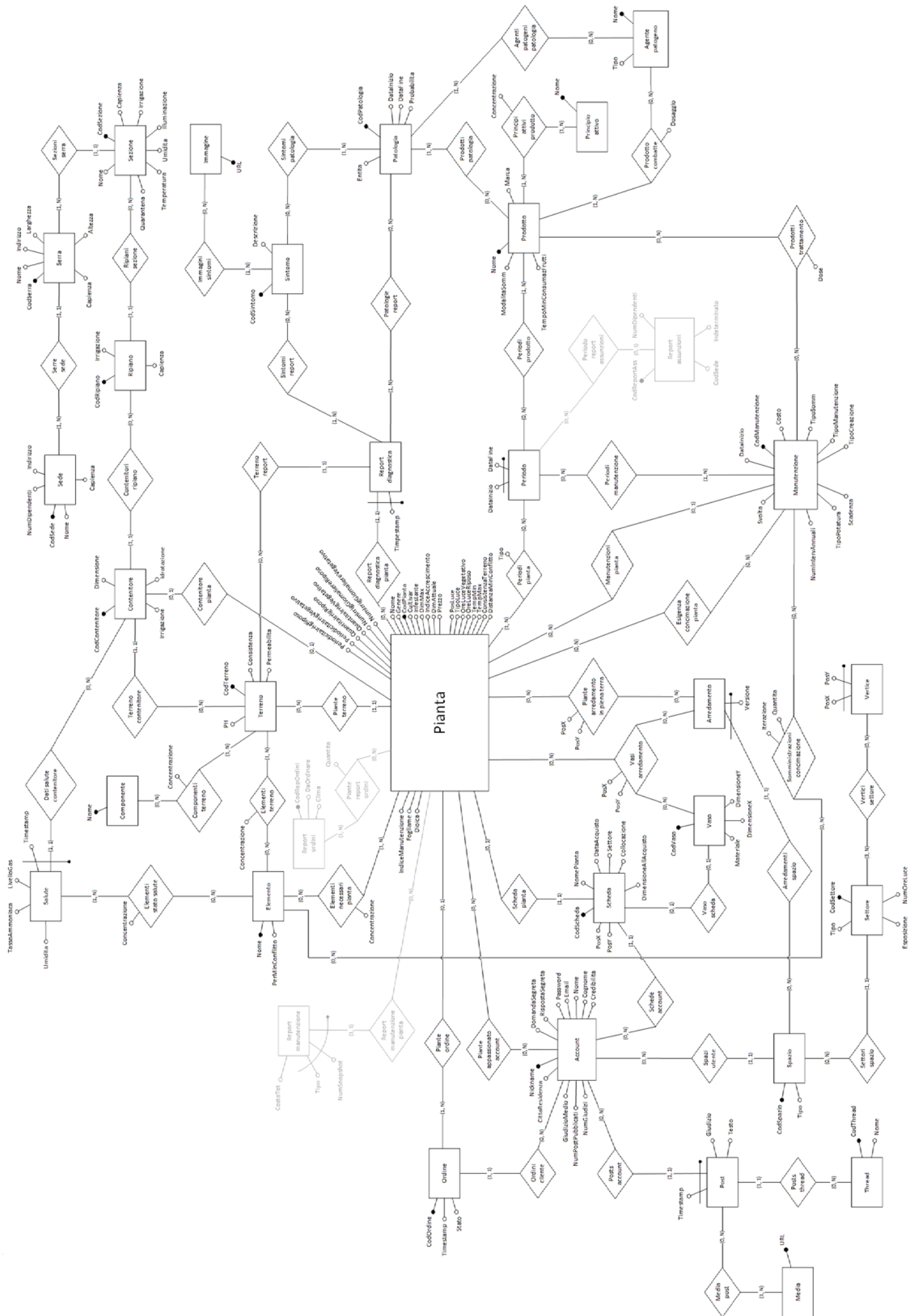
Sarà infine necessario aggiungere un attributo per distinguere le entità figlie: tale attributo sarà *TipoManutenzione*.

La generalizzazione viene quindi così ristrutturata:



3.4 Schema E-R ristrutturato

Lo schema E-R ristrutturato è riportato di seguito:



4. Individuazione di operazioni sui dati

Le frequenze di alcune operazioni si basano su stime presenti nel paragrafo 5.1 “Tavola dei volumi”, dove viene stimato il numero di occorrenze di ciascuna entità e relazione.

Inoltre, d’ora in poi nel testo verrà supposto che l’impresa sia in attività da 30 anni.

- **Operazione 1**
 - Descrizione: Ottenere il numero di posts pubblicati da un account.
 - Frequenza: 1000 volte al giorno. Di solito, nei forum, questo dato è presente sotto il nickname in ogni post pubblicato, perciò è legittimo supporre che sia un’operazione svolta molto frequentemente.
- **Operazione 2**
 - Descrizione: Ottenere il costo complessivo di manutenzione di una specie di pianta.
 - Frequenza: 10 volte al giorno.
- **Operazione 3**
 - Descrizione: Ottenere il numero di piante appartenenti ad una sezione
 - Frequenza: 15 volte al giorno.
- **Operazione 4**
 - Descrizione: Ottenere la specie di pianta più venduta.
 - Frequenza: 20 volte al giorno.
- **Operazione 5**
 - Descrizione: Inserire di una nuova pianta nel magazzino.
 - Frequenza: 4 volte al giorno.
Continuando a supporre che l’impresa sia in attività da 30 anni, sono arrivate mediamente $34560 / 30 = 1152$ piante ogni anno, ovvero 3,16 piante al giorno, approssimato a 4 per un ulteriore margine di sicurezza.
- **Operazione 6**
 - Descrizione: Creare di un nuovo arredamento.
 - Frequenza: 2 volte al giorno.
Continuando a supporre che l’impresa sia in attività da 30 anni, sono stati creati mediamente $13824 / 30 = 460,8$ arredamenti all’anno, ovvero 1,26 arredamenti al giorno. Tale valore viene approssimato a 2 sempre per un ulteriore margine di sicurezza.
- **Operazione 7**
 - Descrizione: Trovare il thread con più contenuti multimediali.
 - Frequenza: 5 volte al giorno.
- **Operazione 8**
 - Descrizione: Ottenere la specie pianta che si ammala più spesso
 - Frequenza: 10 volte al giorno.
Si suppone che questa operazione abbia la stessa frequenza dell’operazione 2.

5. Analisi delle prestazioni delle operazioni

5.1 Tavola dei volumi

Nome	E/R	Numero istanze	Motivazione
Sede	E	20	Ipotesi iniziale.
SerreSede	R	60	In media ogni sede possiede 3 serre. $60 = 20 * 3$
Serra	E	60	Cardinalità (1, 1) con <i>SerreSede</i> .
SezioniSerra	R	480	In media ogni serra possiede 8 sezioni. $480 = 60 * 8$
Sezione	E	480	Cardinalità (1, 1) con <i>SezioniSerra</i> .
RipianiSezione	R	1920	In media ogni sezione possiede 4 ripiani. $1920 = 480 * 4$
Ripiano	E	1920	Cardinalità (1, 1) con <i>RipianiSezione</i> .
ContenitoriRipiano	R	7680	in media ogni ripiano possiede 4 contenitori. $7680 = 1920 * 4$
Contenitore	E	7680	Cardinalità (1, 1) con <i>ContenitoriRipiano</i> .
ContenitorePianta	R	6912	In media 1 contenitore su 10 è vuoto. $6912 = 7680 * 0,9$ Indica inoltre il numero delle piante presenti in magazzino. Ciò implica che in media: <ul style="list-style-type: none"> • Ogni sede contiene $6912 / 20 = 345,6$ piante. • Ogni serra contiene $6912 / 60 = 115,2$ piante. • Ogni sezione possiede $6912 / 480 = 14,4$ piante. • Ogni ripiano possiede $6912 / 1920 = 3,6$ piante.
Pianta	E	34560	In media, nel database, il 20% delle piante è in magazzino. $34560 = 6912 * 5$
PianteTerreno	R	34560	Cardinalità (1, 1) con <i>Pianta</i> .
PeriodiPianta	R	69120	In media, ogni pianta ha due periodi ogni 365 giorni: uno di riposo e uno vegetativo. $69120 = 34560 * 2$
TerrenoContenitore	R	7680	Cardinalità (1, 1) con <i>Contenitore</i> .
SchedaPianta	R	27648	In media, nel database, il 20% delle piante è in magazzino: quindi, ne sono state vendute l'80%. $27648 = 34560 * 0,8$
Scheda	E	27648	Cardinalità (1, 1) con <i>SchedaPianta</i> .
SchedeAccount	R	27648	Cardinalità (1, 1) con <i>Scheda</i> .
Account	E	1152	In media, ogni account possiede 24 schede relative a 24 piante acquistate. $1152 = 27648 / 24$
PianteAppassionatoAccount	R	11520	In media, ogni account è appassionato di 10 piante. $11520 = 1152 * 10$
PostsAccount	R	34560	In media, ogni account pubblica 30 posts. $34560 = 1152 * 30$
Post	E	34560	Cardinalità (1, 1) con <i>PostsAccount</i> .
PostsThread	R	34560	Cardinalità (1, 1) con <i>Post</i> .
Thread	E	864	In media, ogni thread contiene 40 posts. $864 = 34560 / 40$

MediaPost	R	6912	In media, 1 post su 5 contiene un contenuto multimediale. $6912 = 34560 / 5$
Media	E	6480	In media, 1 contenuto multimediale su 16 è duplicato. $6480 = 6912 * (1 - 1 / 16)$
OrdiniCliente	R	5760	In media, ogni account effettua 5 ordini. $5760 = 1152 * 5$
Ordine	E	5760	Cardinalità (1, 1) con <i>OrdiniCliente</i> .
PianteOrdine	R	23040	In media, ogni ordine coinvolge 4 piante. $23040 = 5760 * 4$
SpaziUtente	R	4608	In media, ogni account possiede 2 spazi verdi. $4608 = 1152 * 4$
Spazio	E	4608	Cardinalità (1, 1) con <i>SpaziUtente</i> .
SettoriSpazio	R	18432	In media, ogni spazio contiene 4 settori. $18432 = 4608 * 4$
Settore	E	18432	Cardinalità (1, 1) con <i>SettoriSpazio</i> .
VerticiSettore	R	73728	In media ogni settore ha 4 vertici. $73728 = 18432 * 4$
Vertice	E	73728	Nel peggiore dei casi nessun vertice è ripetuto.
ArredamentiSpazio	R	13824	In media, ogni spazio ha 3 diversi arredamenti. $13824 = 4608 * 3$
Arredamento	E	13824	Cardinalità (1, 1) con <i>ArredamentiSpazio</i> .
VasoScheda	R	13824	In media, metà delle piante vendute è in piena terra, e metà è in un vaso. $13824 = 27648 / 2$
Vaso	E	10368	In media, il 25% dei vasi sono vuoti. $10368 = 13824 * 0,75$
VasiArredamento	R	110592	In media, ogni arredamento possiede 8 vasi. $110592 = 13824 * 8$ Ciò implica che in media, uno stesso vaso è presente in $110592 / 10368 = 10,7$ versioni di arredamento.
PianteArredamentoInPienaTerra	R	165888	In media, ogni arredamento possiede 12 piante in piena terra. $165888 = 13824 * 12$
Patologia	E	100	Ipotesi iniziale.
SintomiPatologia	R	200	In media, ogni patologia presenta 2 sintomi. $200 = 100 * 2$
Sintomo	E	40	Ipotesi iniziale.
ImmaginiSintomi	R	120	In media, ogni sintomo ha 3 immagini rappresentative. $120 = 40 * 3$
Immagine	E	120	Nel peggiore dei casi le immagini rappresentative dei sintomi sono tutte diverse.
ProdottiPatologia	R	150	In media, ogni patologia è combattuta con 1,5 prodotti. $150 = 100 * 1,5$
AgentiPatogeniPatologia	R	150	In media, ogni patologia è causata da 1,5 agenti patogeni. $150 = 100 * 1,5$
Prodotto	E	70	Ipotesi iniziale.
PrincipiAttiviProdotto	R	140	In media, ogni prodotto è composto da 2 principi attivi. $140 = 70 * 2$
PrincipioAttivo	E	50	Ipotesi iniziale.
ProdottoCombatte	R	140	In media, ogni prodotto combatte 2 agenti patogeni. $140 = 70 * 2$

AgentePatogeno	E	90	Ipotesi iniziale.
PeriodiProdotto	R	140	In media, ogni prodotto ha 2 periodi in cui ne è consentito l'utilizzo. $140 = 70 * 2$
Terreno	E	70	Ipotesi iniziale.
ComponentiTerreno	R	175	In media, ogni terreno è composto da 2,5 componenti. $175 = 70 * 2,5$
Componente	E	50	Ipotesi iniziale.
ElementiRerreno	R	210	In media, ogni terreno è composto da 3 elementi. $210 = 70 * 3$
DatiSaluteContenitore	R	38400	In media, ogni contenitore contiene uno storico di 5 dati salute. $38400 = 7680 * 5$
Salute	E	38400	Cardinalità (1, 1) con <i>DatiSaluteContenitore</i> .
ElementiStatoSalute	R	115200	In media, ogni dato relativo ad uno stato di salute contiene 3 elementi. $115200 = 38400 * 3$
Elemento	E	100	Ipotesi iniziale.
ElementiNecessariPianta	R	138240	In media, ogni pianta ha bisogno di 4 elementi per sopravvivere. $138240 = 34560 * 4$
ManutenzioniPianta	R	1728000	In media, ad ogni pianta vengono effettuati 50 interventi di manutenzione nella sua vita. $1728000 = 34560 * 50$
EsigenzaConcimazionePianta	R	69120	In media, ogni pianta richiede 2 tipi di concimazioni necessari per la propria sopravvivenza. $69120 = 34560 * 2$
Manutenzione	E	1762560	Il numero delle sue occorrenze sarà dato dal numero di manutenzioni svolte sulle piante più il numero dei diversi tipi di concimazione richiesti. Ipotizzando che il 50% dei tipi di concimazione richiesti siano duplicati, il numero delle occorrenze dell'entità è dato da: $1762560 = 1728000 + 69120 / 2$
SomministrazioniConcimazione	R	1382400	Tenendo conto che ogni pianta ha bisogno in media di 4 elementi per sopravvivere, supponendo che in media abbia bisogno di 10 somministrazioni per ogni elemento diverso e ricordando che i diversi tipi di concimazione sono $69120 / 2 = 34560$, il numero di occorrenze della relazione è: $1382400 = 69120 / 2 * 4 * 10$
ProdottiTrattamento	R	518400	In media, ogni trattamento richiede l'utilizzo di 1,5 prodotti. Supponendo che i trattamenti siano il 20% delle manutenzioni svolte: $518400 = 1728000 * 0,2 * 1,5$
PeriodiManutenzione	R	3525120	In media, ogni manutenzione ha 2 periodi in cui essa è consentita. $3525120 = 1762560 * 2$
ReportManutenzione	E	4147200	Supponendo che l'impresa sia in attività da 30 anni, essendo compilato un report per ogni pianta ogni 3 mesi: $4147200 = 30 * 12 / 3 * 34560$
ReportManutenzionePianta	R	4147200	Cardinalità (1, 1) con <i>ReportManutenzione</i> .

ReportOrdini	E	1560	Supponendo che l'impresa sia in attività da 30 anni, essendo compilato un report degli ordini ogni settimana: $1560 = 30 * 52$
PianteReportOrdini	R	15600	In media, ogni report degli ordini contiene 10 piante. $15600 = 1560 * 10$
ReportAssunzioni	E	360	Supponendo che l'impresa sia in attività da 30 anni, essendo compilato un report delle assunzioni ogni mese: $360 = 30 * 12$
PeriodoReportAssunzioni	R	288	In media, 8 report di assunzioni su 10 prevedono assunzioni a tempo determinato: $288 = 360 * 0,8$
Periodo	E	2184620	Supponendo che il 40% dei periodi delle manutenzioni siano duplicati, si ha: $2184620 = 3525120 * 0,6 + 69120 + 140 + 288$
ReportDiagnosticaPianta	R	103680	In media, ogni pianta si ammala 3 volte durante la sua vita. $103680 = 34560 * 3$
ReportDiagnostica	E	103680	Cardinalità (1, 1) con <i>ReportDiagnosticaPianta</i> .
PatologieReport	R	124416	In media, ad ogni rivelamento di patologia, ogni pianta si ammala di 1,2 patologie. $124416 = 103680 * 1,2$
TerrenoReport	R	103680	Cardinalità (1, 1) con <i>ReportDiagnostica</i> .
SintomiReport	R	207360	In media, ad ogni rivelamento di patologia, ogni pianta presenta 2 sintomi. $207360 = 103680 * 2$

5.2 Tavole degli accessi

5.2.1 Operazione 1: Ottenere il numero di posts pubblicati da un account

- **Descrizione dell'operazione**

Ottiene il numero esatto di post pubblicati sul forum da parte di un *Account*.

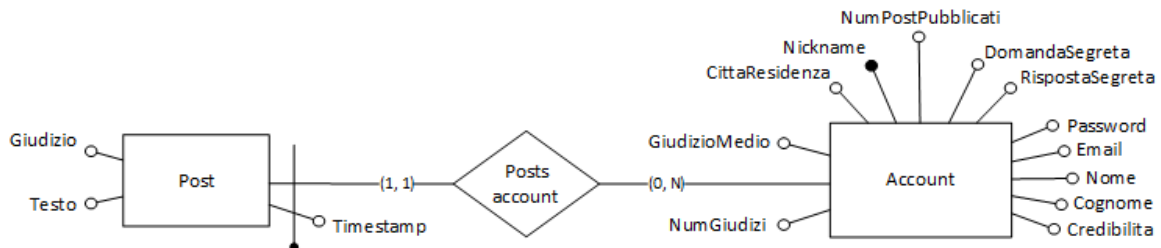
- **Input**

- *Nickname* dell'*Account* del quale si desidera trovare il numero di posts pubblicati.

- **Output**

- Numero di posts pubblicati dall'utente in questione.

- **Porzione del diagramma E-R interessata**



- **Porzione della tavola dei volumi interessata**

Nome	E/R	Numero istanze	Motivazione
Account	E	1152	In media, ogni account possiede 24 schede relative a 24 piante acquistate. $1152 = 27648 / 24$
PostsAccount	R	34560	In media, ogni account pubblica 30 posts. $34560 = 1152 * 30$
Post	E	34560	Cardinalità (1, 1) con <i>PostsAccount</i> .

- **Tavola degli accessi**

Senza Ridondanza 5				
Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
34560	Letture	E	Post	Vengono lette tutte le istanze di <i>Post</i> per recuperare il numero di posts pubblicati da un <i>Account</i> .
34560	Totale operazioni elementari per singola operazione.			
$34560 * 1000 = 34560000$	Totale operazioni elementari al giorno.			

Con Ridondanza 5				
Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
1	Letture	E	Account	Viene compiuta 1 operazione di lettura in <i>Account</i> per ottenere il numero di <i>Post</i> pubblicati da un utente.
1	Totale operazioni elementari per singola operazione.			
$1 * 1000 = 1000$	Totale operazioni elementari al giorno.			

5.2.2 Operazione 2: Ottenere il costo complessivo di manutenzione di una specie di pianta

- **Descrizione dell'operazione**

Ottiene il costo complessivo di tutte le manutenzioni effettuate su tutte le piante di una certa specie effettuate fino al momento dell'esecuzione dell'operazione.

Questa operazione è utile al fine di compilare i *ReportManutenzione* e anche durante la visualizzazione di piante suggerite per l'arredo al variare del grado medio di manutenzione scelto dal cliente durante l'arredo.

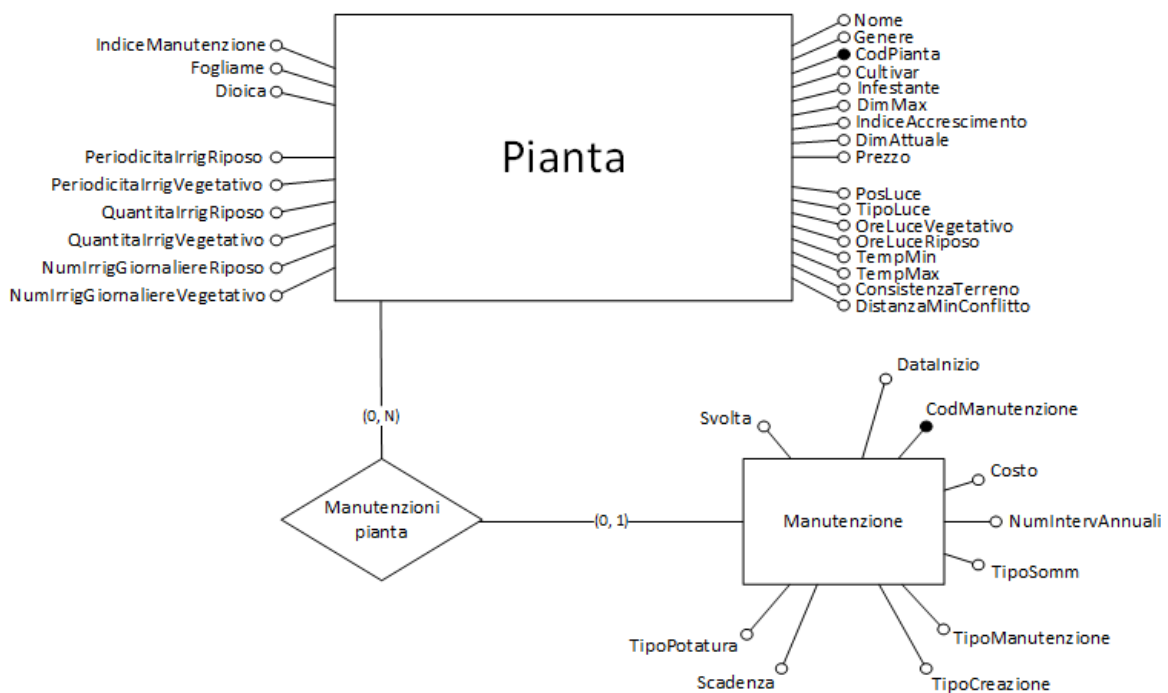
- **Input**

- Nome della specie di *Pianta* della quale si desidera trovare il costo complessivo di manutenzione.

- **Output**

- Costo complessivo di tutte le manutenzioni effettuate su una specie di *Pianta*.

- **Porzione del diagramma E-R interessata**



- **Porzione della tavola dei volumi interessata**

Nome	E/R	Numero istanze	Motivazione
Pianta	E	34560	In media, nel database, il 20% delle piante è in magazzino. $34560 = 6912 * 5$
ManutenzioniPianta	R	1728000	In media, ad ogni pianta vengono effettuati 50 interventi di manutenzione nella sua vita. $1728000 = 34560 * 50$
Manutenzione	E	1762560	Il numero delle sue occorrenze sarà dato dal numero di manutenzioni svolte sulle piante più il numero dei diversi tipi di concimazione richiesti. Ipotizzando che il 50% dei tipi di concimazione richiesti siano duplicati, il numero delle occorrenze dell'entità è dato da: $1762560 = 1728000 + 69120 / 2$

- **Tavola degli accessi**

Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
34560	Lettura	E	Pianta	Lettura di tutte le istanze di <i>Pianta</i> per trovare i codici relativi a tutti gli esemplari di una specie.
$50 * 34560 / 2 = 864000$	Lettura	E	Manutenzione	Lettura dei costi delle manutenzioni relative alla specie della pianta in questione. Si considera improbabile che più di metà delle occorrenze di <i>Pianta</i> siano appartenenti alla stessa specie.
864000	Totale operazioni elementari per singola operazione.			
$864000 * 10 = 8640000$	Totale operazioni elementari al giorno.			

5.2.3 Operazione 3: Ottenere il numero di piante appartenenti ad una sezione

- **Descrizione dell'operazione**

Ottiene il numero di piante presenti in una determinata sezione aziendale.

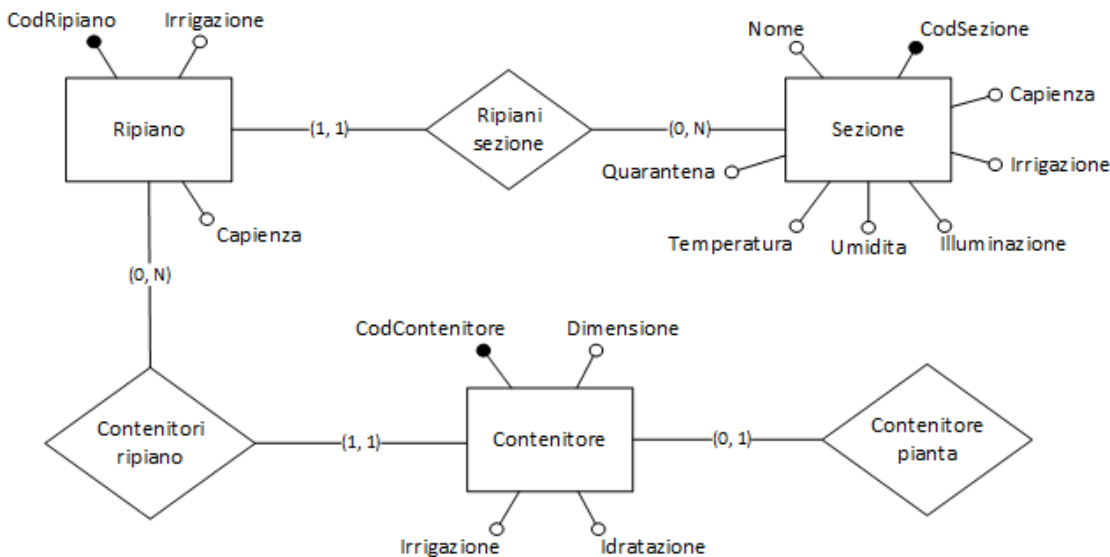
- **Input**

- *CodSezione* di *Sezione*.

- **Output**

- Numero di piante presenti nella *Sezione* con codice *CodSezione*.

- **Porzione del diagramma E-R interessata**



- **Porzione della tavola dei volumi interessata**

Nome	E/R	Numero istanze	Motivazione
Sezione	E	480	Cardinalità (1, 1) con <i>SezioniSerra</i> .
RipianiSezione	R	1920	In media ogni sezione possiede 4 ripiani. $1920 = 480 * 4$
Ripiano	E	1920	Cardinalità (1, 1) con <i>RipianiSezione</i> .
ContenitoriRipiano	R	7680	in media ogni ripiano possiede 4 contenitori. $7680 = 1920 * 4$
Contenitore	E	7680	Cardinalità (1, 1) con <i>ContenitoriRipiano</i> .
ContenitorePianta	R	6912	In media 1 contenitore su 10 è vuoto. $6912 = 7680 * 0,9$ Indica inoltre il numero delle piante presenti in magazzino. Ciò implica che in media: <ul style="list-style-type: none"> • Ogni sede contiene $6912 / 20 = 345,6$ piante. • Ogni serra contiene $6912 / 60 = 115,2$ piante. • Ogni sezione possiede $6912 / 480 = 14,4$ piante. Ogni ripiano possiede $6912 / 1920 = 3,6$ piante.

- **Tavola degli accessi**

Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
1920	Lettura	E	Ripiano	Lettura di <i>CodSezione</i> di tutte le occorrenze dell'entità <i>Ripiano</i> , per trovare i ripiani appartenenti alla sezione di cui si desidera trovare il numero di piante appartenenti.
7680	Lettura	E	Contenitore	Lettura di <i>CodPianta</i> di tutte le occorrenze dell'entità <i>Contenitore</i> che abbiano come <i>CodRipiano</i> uno dei ripiani trovati precedentemente.
9600	Totale operazioni elementari per singola operazione.			
$9600 * 15 = 144000$	Totale operazioni elementari al giorno.			

5.2.4 Operazione 4: Ottenere la specie di pianta più venduta

- **Descrizione dell'operazione**

Ottiene il nome della specie di *Pianta* più venduta, ovvero che la specie che partecipa di più nella relazione *SchedaPianta*.

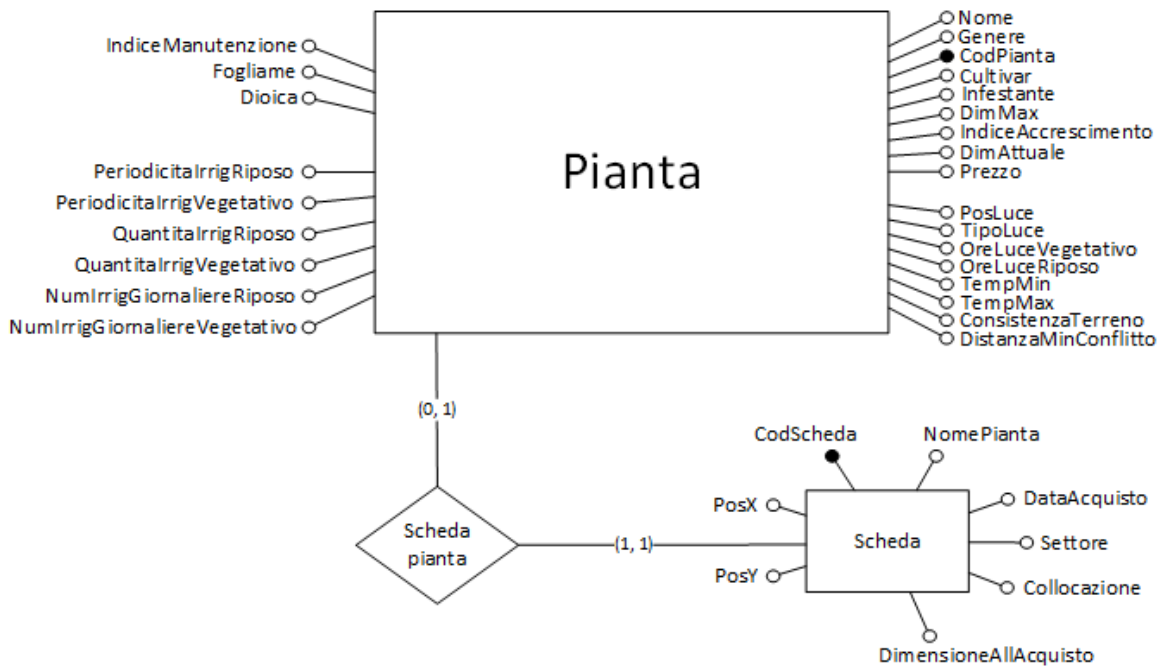
- **Input**

Nessuno

- **Output**

- Nome della *Pianta* più venduta fino al momento dell'esecuzione dell'operazione.

- **Porzione del diagramma E-R interessata**



- **Porzione della tavola dei volumi interessata**

Nome	E/R	Numero istanze	Motivazione
Pianta	E	34560	In media, nel database, il 20% delle piante è in magazzino. $34560 = 6912 * 5$
SchedaPianta	R	27648	In media, nel database, il 20% delle piante è in magazzino: quindi, ne sono state vendute l'80%. $27648 = 34560 * 0,8$
Scheda	E	27648	Cardinalità (1, 1) con <i>SchedaPianta</i> .

- **Tavola degli accessi**

Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
27648	Lettura	E	Scheda	Lettura di <i>CodPianta</i> di tutte le istanze di <i>Scheda</i> .
27648	Lettura	E	Pianta	Lettura di <i>Nome</i> di tutte le istanze di <i>Pianta</i> collegate alle istanze di <i>Scheda</i> .
55296	Totale operazioni elementari per singola operazione.			
$55296 * 20 = 1105920$	Totale operazioni elementari al giorno.			

5.2.5 Operazione 5: Inserire di una nuova pianta nel magazzino

- **Descrizione dell'operazione**

Inserisce un nuovo esemplare di *Pianta* da inserire in un opportuno *Contenitore* nel magazzino. Risulta utile nel momento in cui arrivano nuovi esemplari dai fornitori.

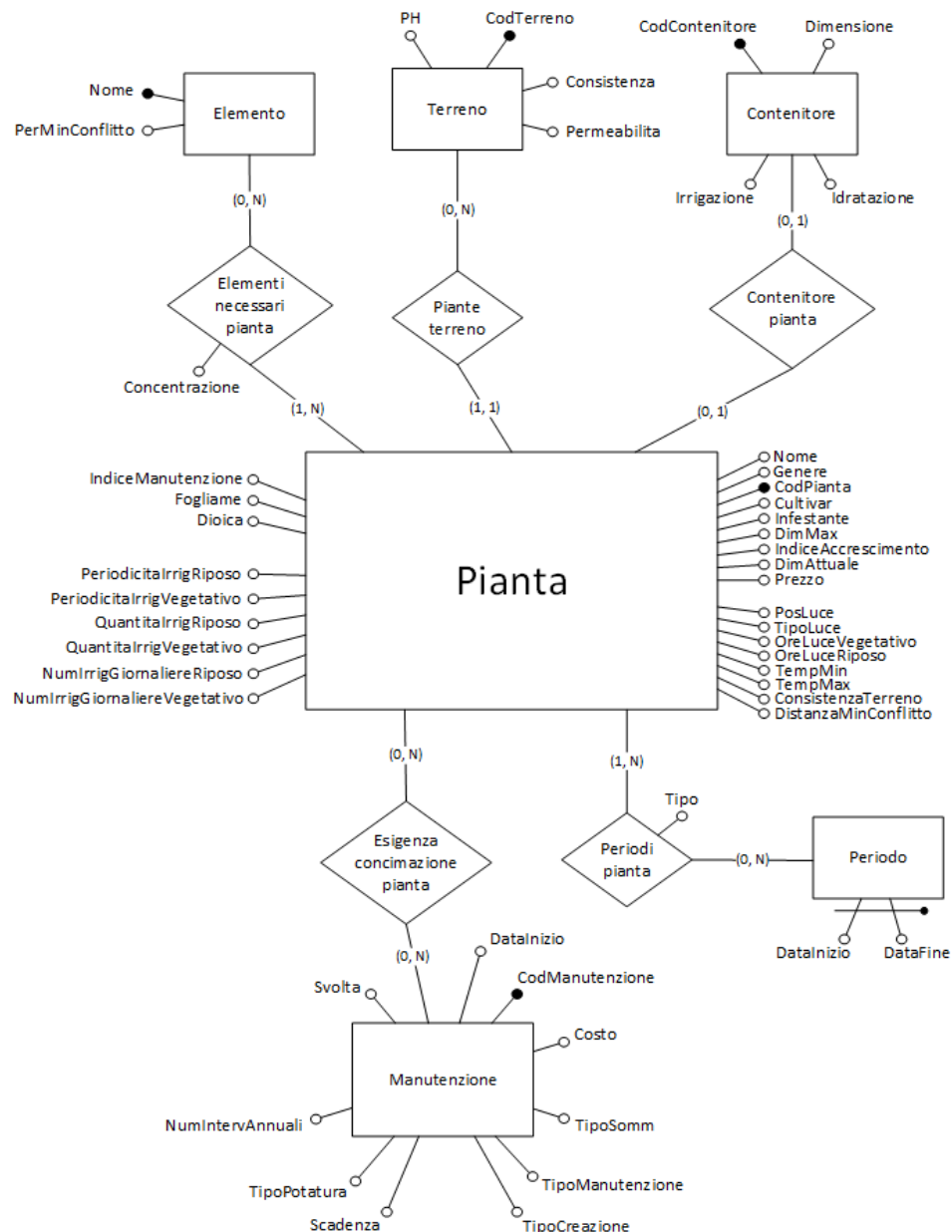
- **Input**

- Un'occorrenza dell'entità *Pianta*.
- *CodContenitore* in cui inserire la *Pianta*.
- Occorrenze dell'entità *Periodo* relative ai periodi vegetativi e/o di riposo dell'esemplare.
- Occorrenze dell'entità *Manutenzione* relative alle concimazioni richieste dalla *Pianta* per la propria sopravvivenza.
- Occorrenze dell'entità *Elemento* relative agli elementi necessari dalla *Pianta* per la propria sopravvivenza.

- **Output**

Nessuno.

- **Porzione del diagramma E-R interessata**



- **Porzione della tavola dei volumi interessata**

Nome	E/R	Numero istanze	Motivazione
Pianta	E	34560	In media, nel database, il 20% delle piante è in magazzino. $34560 = 6912 * 5$
PeriodiPianta	R	69120	In media, ogni pianta ha due periodi ogni 365 giorni: uno di riposo e uno vegetativo. $69120 = 34560 * 2$
EsigenzaConcimazionePianta	R	69120	In media, ogni pianta richiede 2 tipi di concimazioni necessari per la propria sopravvivenza. $69120 = 34560 * 2$
ElementiNecessariPianta	R	138240	In media, ogni pianta ha bisogno di 4 elementi per sopravvivere. $138240 = 34560 * 4$

- **Tavola degli accessi**

Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costrutto	Descrizione
$1 * 2 = 2$	Scrittura	E	Pianta	1 operazione di scrittura su <i>Pianta</i> per inserirne una nuova enupla (inclusi <i>CodContentore</i> e <i>CodTerreno</i>).
$4 * 2 = 8$	Scrittura	R	ElementiNecessariPianta	4 operazioni di scrittura su <i>ElementiNecessariPianta</i> per associare all'esemplare gli elementi di cui ha bisogno.
$2 * 2 = 4$	Scrittura	R	PeriodiPianta	2 operazioni di scrittura su <i>PeriodiPianta</i> per associare alla pianta i periodi vegetativi e/ di riposo.
$2 * 2 = 4$	Scrittura	R	EsigenzaConcimazionePianta	2 operazioni di scrittura su <i>EsigenzaConcimazionePianta</i> per associare alla <i>Pianta</i> le concimazioni che necessita.
18	Totale operazioni elementari per singola operazione.			
$18 * 4 = 72$	Totale operazioni elementari al giorno.			

5.2.6 Operazione 6: Creare un nuovo arredamento

- **Descrizione dell'operazione**

Creare una nuova versione di arredamento relativo ad uno *Spazio* di un utente.

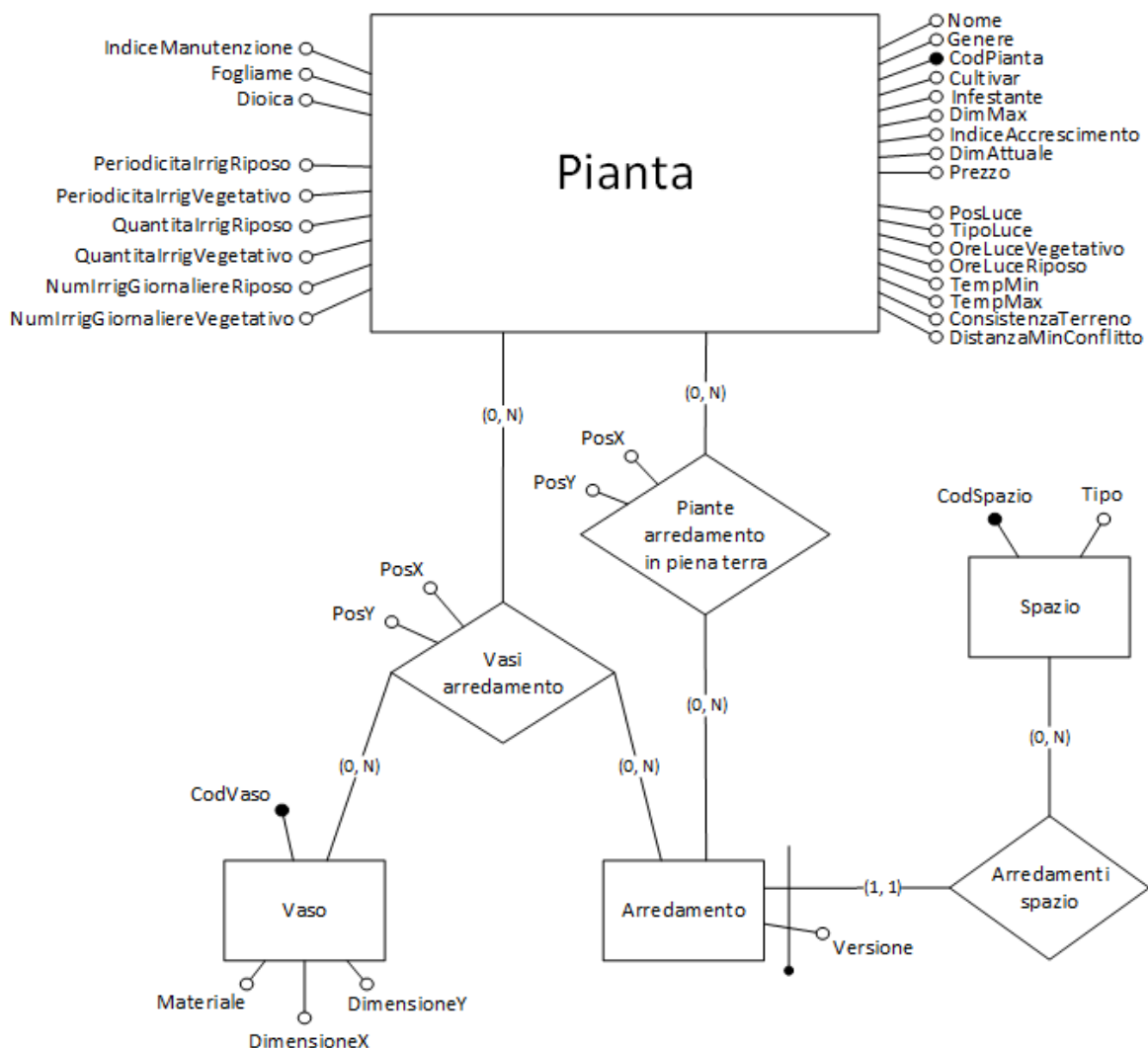
- **Input**

- Occorrenza di *Arredamento* (ovvero la *Versione* e il *CodSpazio* associato).
- Occorrenze di *VasiArredamento* per associare all'arredamento gli esemplari contenuti in un vaso.
 - Occorrenze delle entità *Vaso* e *Pianta* per ogni associazione.
- Occorrenze di *PianteArredamentoInPienaTerra* per associare all'arredamento gli esemplari piantati direttamente nel terreno.
 - Occorrenza dell'entità *Pianta* per ogni associazione.

- **Output**

Nessuno.

- **Porzione del diagramma E-R interessata**



- **Porzione della tavola dei volumi interessata**

Nome	E/R	Numero istanze	Motivazione
Pianta	E	34560	In media, nel database, il 20% delle piante è in magazzino. $34560 = 6912 * 5$
VasiArredamento	R	110592	In media, ogni arredamento possiede 8 vasi. $110592 = 13824 * 8$ Ciò implica che in media, uno stesso vaso è presente in $110592 / 10368 = 10,7$ versioni di arredamento.
PianteArredamentoInPienaTerra	R	165888	In media, ogni arredamento possiede 12 piante in piena terra. $165888 = 13824 * 12$
Vaso	E	10368	In media, il 25% dei vasi sono vuoti. $10368 = 13824 * 0,75$
ArredamentiSpazio	R	13824	In media, ogni spazio ha 3 diversi arredamenti. $13824 = 4608 * 3$
Arredamento	E	13824	Cardinalità (1, 1) con <i>ArredamentiSpazio</i> .
Spazio	E	4608	Cardinalità (1, 1) con <i>SpaziUtente</i> .

- **Tavola degli accessi**

Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
$1 * 2 = 2$	Scrittura	E	Arredamento	1 operazione di scrittura su <i>Arredamento</i> per inserire una nuova enupla (incluso <i>CodSpazio</i>).
$12 * 2 = 24$	Scrittura	R	PianteArredamentoInPiena Terra	12 operazioni di scrittura su <i>PianteArredamentoInPienaTerra</i> per associare all'arredamento le piante piantate nella piena terra.
$8 * 2 = 16$	Scrittura	R	VasiArredamento	8 operazioni di scrittura su <i>VasiArredamento</i> per associare all'arredamento le piante contenute in un vaso.
42	Totale operazioni elementari per singola operazione.			
$42 * 2 = 84$	Totale operazioni elementari al giorno.			

5.2.7 Operazione 7: Trovare il thread con più contenuti multimediali

- **Descrizione dell'operazione**

Trova il *Thread* nel forum che contiene più contenuti multimediali.

Può essere utile per delle funzioni web che creano una galleria delle piante grazie ai contributi multimediali postati dagli utenti sul forum o in fase di eliminazione di threads per liberare memoria.

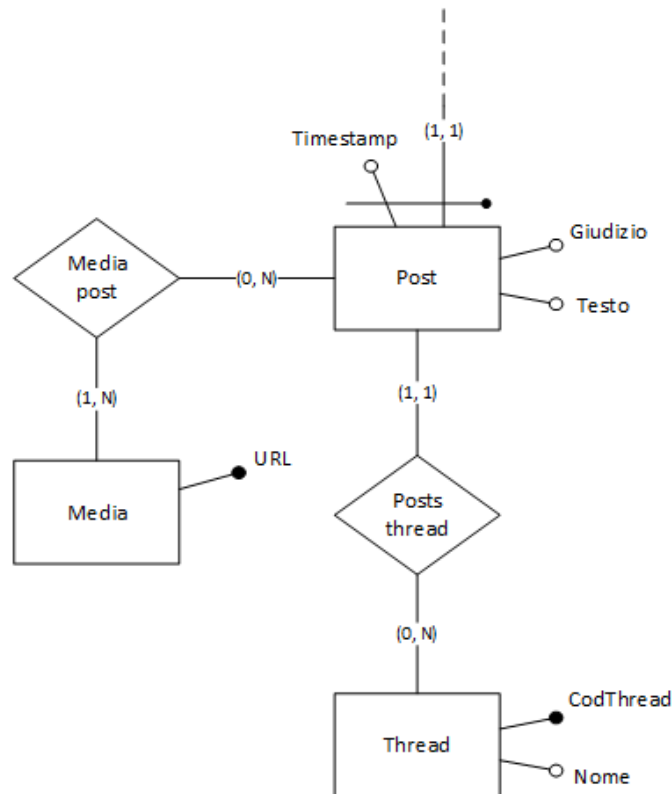
- **Input**

Nessuno.

- **Output**

- *CodThread* del *Thread*

- **Porzione del diagramma E-R interessata**



- **Porzione della tavola dei volumi interessata**

Nome	E/R	Numero istanze	Motivazione
Post	E	34560	Cardinalità (1, 1) con <i>PostsAccount</i> .
PostsThread	R	34560	Cardinalità (1, 1) con <i>Post</i> .
Thread	E	864	In media, ogni thread contiene 40 posts. $864 = 34560 / 40$
MediaPost	R	6912	In media, 1 post su 5 contiene un contenuto multimediale. $6912 = 34560 / 5$
Media	E	6480	In media, 1 contenuto multimediale su 16 è duplicato. $6480 = 6912 * (1 - 1 / 16)$

- **Tavola degli accessi**

Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
6912	Lettura	R	MediaPost	Lettura di tutte le istanze della relazione <i>MediaPost</i> per trovare i post relativi ai media pubblicati.
6912	Lettura	E	Post	Lettura dell'attributo <i>CodThread</i> di <i>Post</i> per apprendere in che <i>Thread</i> il <i>Post</i> in questione appartiene.
13824	Totale operazioni elementari per singola operazione.			
13824 * 5 = 69120	Totale operazioni elementari al giorno.			

5.2.8 Operazione 8: Ottenere la specie pianta che si ammala più spesso

- **Descrizione dell'operazione**

Ottiene la specie di pianta più tendente ad ammalarsi.

Anche questa operazione può essere utile durante la visualizzazione di piante suggerite per l'arredo al variare del grado medio di manutenzione scelto dal cliente durante l'arredo.

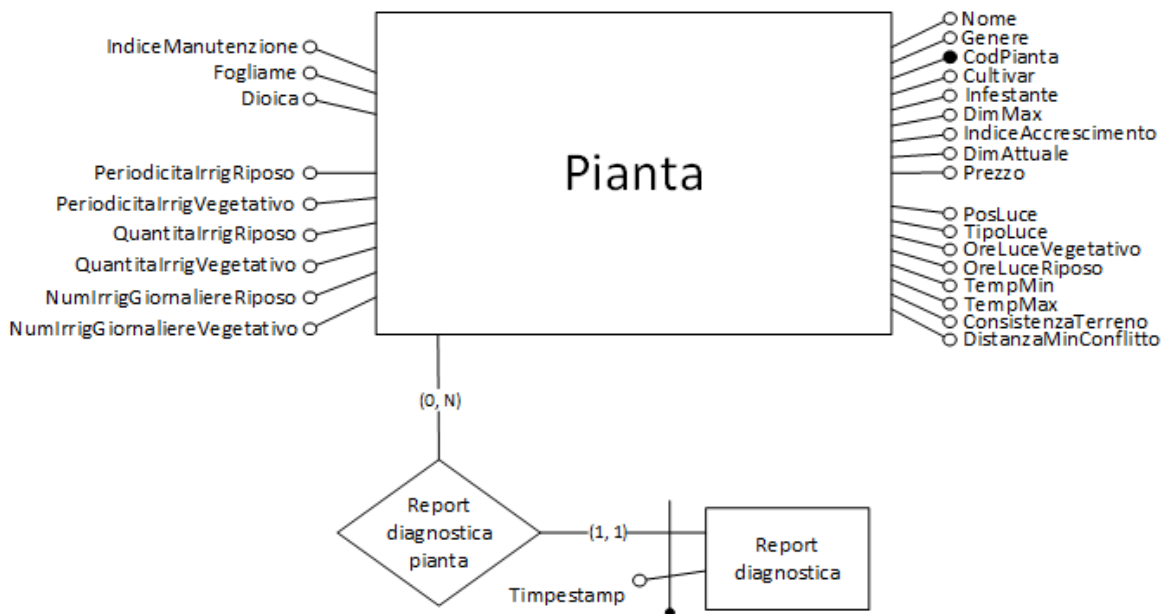
- **Input**

Nessuno.

- **Output**

- Nome della specie di *Pianta* che più spesso si ammala.

- **Porzione del diagramma E-R interessata**



- **Porzione della tavola dei volumi interessata**

Nome	E/R	Numero istanze	Motivazione
Pianta	E	34560	In media, nel database, il 20% delle piante è in magazzino. $34560 = 6912 * 5$
ReportDiagnosticaPianta	R	103680	In media, ogni pianta si ammala 3 volte durante la sua vita. $103680 = 34560 * 3$
ReportDiagnostica	E	103680	Cardinalità (1, 1) con <i>ReportDiagnosticaPianta</i> .

- **Tavola degli accessi**

Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
103680	Lettura	E	ReportDiagnostica	Lettura di <i>CodPianta</i> tutte le istanze di <i>ReportDiagnostica</i> per determinare la <i>Pianta</i> che più spesso si ammala.
103680	Lettura	E	Pianta	Lettura di <i>Nome</i> di <i>Pianta</i> per tutte le istanze presenti della relazione <i>ReportDiagnostica</i> .
207360	Totale operazioni elementari per singola operazione.			
$207360 * 10 = 2073600$	Totale operazioni elementari al giorno.			

5.3 Considerazione sulle ridondanze già presenti

L'unica ridondanza già presente che migliora l'efficienza di un'operazione è la numero 4 (l'attributo *NumPostPubblicati* dell'entità *Account*).

Per effettuare la decisione riguardo il suo mantenimento, occorre valutare il costo del suo mantenimento analizzando la tabella degli accessi relativa all'operazione di pubblicazione di un nuovo post da parte di un utente. Considerando che in 30 anni di attività dell'impresa sono stati creati 34560 posts nel forum, sono stati creati in media $34560 / 30 = 1152$ posts all'anno, ovvero in media 3,16 posts al giorno. La frequenza di questa operazione viene arrotondata a 4 per un maggior margine di sicurezza.

Nel peggiore dei casi, il post contiene un contenuto multimediale nuovo e appartiene ad un nuovo *Thread*:

Senza ridondanza				
Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
$1 * 2 = 2$	Scrittura	E	Thread	Creazione di una nuova istanza di <i>Thread</i> .
$1 * 2 = 2$	Scrittura	E	Post	Creazione di una nuova istanza di <i>Post</i> .
$1 * 2 = 2$	Scrittura	E	Media	Creazione di una nuova istanza di <i>Media</i> .
$1 * 2 = 2$	Scrittura	R	MediaPost	Associazione tra il <i>Post</i> appena creato e il contenuto multimediale in esso contenuto.
6	Totale operazioni elementari per singola operazione.			
$6 * 4 = 24$	Totale operazioni elementari al giorno.			

Con ridondanza				
Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
1 * 2 = 2	Scrittura	E	Thread	Creazione di una nuova istanza di <i>Thread</i> .
1 * 2 = 2	Scrittura	E	Post	Creazione di una nuova istanza di <i>Post</i> .
1 * 2 = 2	Scrittura	E	Media	Creazione di una nuova istanza di <i>Media</i> .
1 * 2 = 2	Scrittura	R	MediaPost	Associazione tra il <i>Post</i> appena creato e il contenuto multimediale in esso contenuto.
1 * 2 = 2	Scrittura	E	Account	Aggiornamento della ridondanza <i>NumPostPubblicati</i> nella relativa istanza di <i>Account</i> dell'utente che ha pubblicato il nuovo post.
10	Totale operazioni elementari per singola operazione.			
10 * 4 = 40	Totale operazioni elementari al giorno.			

La ridondanza semplifica notevolmente l'operazione 1, portandola le operazioni elementari giornaliere in media da $34560000 + 24 = 34560024$ a $1000 + 40 = 1040$. È conveniente perciò mantenere tale ridondanza. Tutte le altre ridondanze non influenzano in alcun modo le operazioni rimanenti e non è necessario mantenerle.

6. Introduzione di ridondanze

Per rendere più efficienti le operazioni sopra indicate, verrà proposto l'inserimento di alcune ridondanze, le quali andranno messe in discussione singolarmente per decidere sul loro mantenimento.

6.1 Ridondanza per l'operazione 1

Già presente: si tratta della Ridondanza 5 analizzata nel paragrafo 3.1.3 "Attributi derivabili dal conteggio delle occorrenze", ovvero dell'attributo *NumPostPubblicati* dell'entità *Account*, che semplifica enormemente l'operazione 1.

6.2 Ridondanza per l'operazione 2

6.2.1 Primo tentativo

Uno primo tentativo di ridondanza per l'operazione numero 2 potrebbe consistere nell'inserimento di un nuovo attributo *SpeciePianta* nell'entità *Manutenzione*.

Di seguito, il confronto delle tavole degli accessi con o senza ridondanza in questione:

Senza ridondanza				
Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
34560	Lettura	E	Pianta	Lettura di tutte le istanze di <i>Pianta</i> per trovare i codici relativi a tutti gli esemplari di una specie.
$50 * 34560 / 2 = 864000$	Lettura	E	Manutenzione	Lettura dei costi delle manutenzioni relative alla specie della pianta in questione. Si considera improbabile che più di metà delle occorrenze di <i>Pianta</i> siano appartenenti alla stessa specie.
864000	Totale operazioni elementari per singola operazione.			
$864000 * 10 = 8640000$	Totale operazioni elementari al giorno.			

Con ridondanza				
Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
1762560	Lettura	E	Manutenzione	Lettura di tutte le istanze di <i>Manutenzione</i> per ottenere tutti i costi di manutenzione di una determinata specie di <i>Pianta</i> .
1762560	Totale operazioni elementari per singola operazione.			
1762560 * 10 = 17625600	Totale operazioni elementari al giorno.			

Il numero delle operazioni elementari giornaliere aumenta da 8640000 a 17625600, perciò questa scelta di introduzione di ridondanza non conviene.

Inutile quindi analizzare i costi che introdurrebbe questa ridondanza se introdotta.

6.2.2 Secondo tentativo

Un intervento molto più invasivo sullo schema E-R volto a migliorare notevolmente l'operazione 2 è dato dall'inserimento di una nuova entità e di una nuova relazione.

Si tratta dell'entità *Specie* e della relazione *SpeciePianta*.

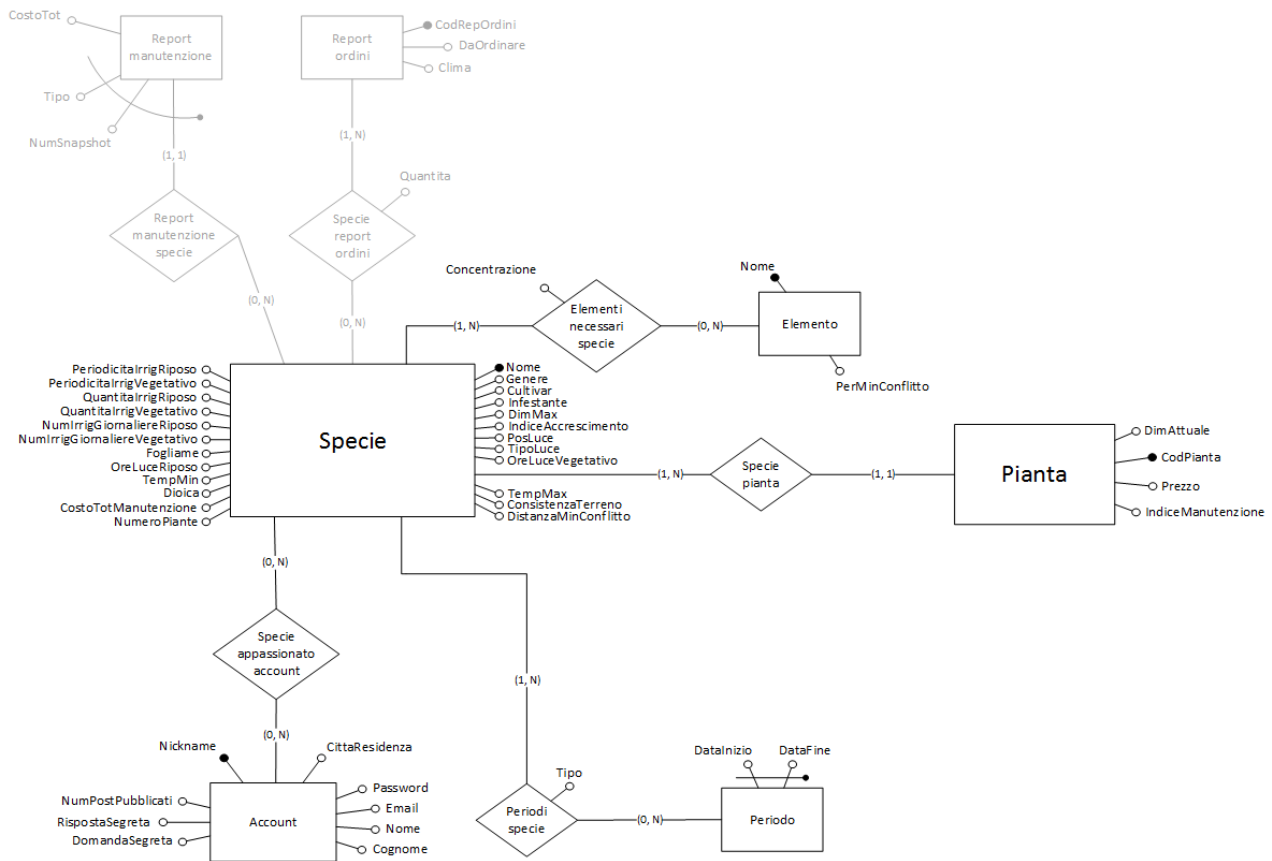
- L'entità *Specie* rappresenta, con un'unica istanza, tutte le piante relative ad una specie. Questa entità racchiude quindi tutte le informazioni a comune tra le piante di una stessa specie. Per migliorare l'efficienza della seconda operazione, si aggiungerà un attributo a tale entità, di nome *CostoTotManutenzione*, il cui valore rappresenta la spesa complessiva di manutenzione su tutte le piante della specie in questione. Per fornire al cliente il costo medio di manutenzione, è conveniente inserire un secondo attributo a *Specie* di nome *NumeroPiante*, il cui valore indica il numero di piante presenti nel database appartenenti alla specie rappresentata dall'istanza. Questo evita la lettura di tutte le istanze dell'entità *Pianta*, equivalente a 34560 operazioni elementari.
- La relazione *SpeciePianta* associa ad ogni pianta presente nel database la sua specie.

Questa modifica altera inevitabilmente il nome di altre 5 relazioni, in modo da mantenere lo standard adottato nella nominazione di tutte le relazioni presenti nello schema E-R.

Più in dettaglio, le relazioni da rinominare sono quelle relative alla *Specie* anziché ad una singola *Pianta*, cioè le seguenti:

- *PianteReportOrdini* diventa *SpecieReportOrdini*;
- *ElementiNecessariPianta* diventa *ElementiNecessariSpecie*;
- *ReportManutenzionePianta* diventa *ReportManutenzioneSpecie*;
- *PianteAppassionatoAccount* diventa *SpecieAppassionatoAccount*;
- *PeriodiPianta* diventa *PeriodiSpecie*.

La porzione di diagramma E-R interessata cambia nel seguente modo:



La tavola dei volumi viene così aggiornata:

Nome	E/R	Numero istanze	Motivazione
SpeciePianta	R	34560	Cardinalità (1, 1) con <i>Pianta</i> .
Specie	E	90	In media, esistono 384 esemplari per ogni <i>Specie</i> . $90 = 34560 / 384$

Di seguito, il confronto delle tavole degli accessi con o senza ridondanza in questione:

Senza ridondanza				
Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
34560	Letture	E	Pianta	Letture di tutte le istanze di <i>Pianta</i> per trovare i codici relativi a tutti gli esemplari di una specie.
$50 * 34560 / 2 = 864000$	Letture	E	Manutenzione	Letture dei costi delle manutenzioni relative alla specie della pianta in questione. Si considera improbabile che più di metà delle occorrenze di <i>Pianta</i> siano appartenenti alla stessa specie.
864000	Totale operazioni elementari per singola operazione.			
$864000 * 10 = 8640000$	Totale operazioni elementari al giorno.			

Con ridondanza				
Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
1	Lettura	E	Specie	Lettura di <i>CostoTotManutenzione</i> dell'entità <i>Specie</i> per ricavare il costo complessivo di manutenzione di una <i>Specie</i> .
1	Totale operazioni elementari per singola operazione.			
1 * 10 = 10	Totale operazioni elementari al giorno.			

Il numero di operazioni elementari giornaliere scende da 8640000 a 10.

Per analizzare il costo di aggiornamento di tale ridondanza, occorre esaminare la tavola degli accessi relativa all'operazione di inserimento di una nuova manutenzione.

Consultando la tavola dei volumi, si nota come in 30 anni di attività dell'impresa siano state effettuate 1728000 manutenzioni, il che implica che in media sono state effettuate $1728000 / 30 = 57600$ manutenzioni all'anno, ovvero in media $57600 / 365 = 158$ manutenzioni al giorno.

Di seguito, il confronto delle tavole di accesso relativo a tale inserimento, con o senza ridondanza:

Senza ridondanza				
Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
1 * 2 = 2	Scrittura	E	Manutenzione	Creazione di una nuova istanza dell'entità <i>Manutenzione</i> .
2 * 2 = 4	Scrittura	E	Periodo	Creazione di due nuove istanze dell'entità <i>Periodo</i> relative ai periodi durante i quali tale manutenzione è consentita.
6	Totale operazioni elementari per singola operazione.			
6 * 158 = 948	Totale operazioni elementari al giorno.			

Con ridondanza				
Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
1 * 2 = 2	Scrittura	E	Manutenzione	Creazione di una nuova istanza dell'entità <i>Manutenzione</i> .
2 * 2 = 4	Scrittura	E	Periodo	Creazione di due nuove istanze dell'entità <i>Periodo</i> relative ai periodi durante i quali tale manutenzione è consentita.
1	Lettura	E	Pianta	Lettura dell'entità <i>Pianta</i> per trovarne la specie.
1 * 2 = 2	Scrittura	E	Specie	Aggiornamento dell'attributo <i>CostoTotManutenzione</i> di <i>Specie</i> .
9	Totale operazioni elementari per singola operazione.			
9 * 158 = 1422	Totale operazioni elementari al giorno.			

Si rende inoltre necessario aggiornare l'attributo *NumeroPiante* dell'entità *Specie* durante l'inserimento di una nuova Pianta (operazione 5).

Senza ridondanza				
Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costrutto	Descrizione
1 * 2 = 2	Scrittura	E	Pianta	1 operazione di scrittura su <i>Pianta</i> per inserirne una nuova ennupla (inclusi <i>CodContenitore</i> e <i>CodTerreno</i>).
4 * 2 = 8	Scrittura	R	ElementiNecessariPianta	4 operazioni di scrittura su <i>ElementiNecessariPianta</i> per associare all'esemplare gli elementi di cui ha bisogno.
2 * 2 = 4	Scrittura	R	PeriodiPianta	2 operazioni di scrittura su <i>PeriodiPianta</i> per associare alla pianta i periodi vegetativi e/ di riposo.
2 * 2 = 4	Scrittura	R	EsigenzaConcimazionePianta	2 operazioni di scrittura su <i>EsigenzaConcimazionePianta</i> per associare alla <i>Pianta</i> le concimazioni che necessita.
18	Totale operazioni elementari per singola operazione.			
18 * 4 = 72	Totale operazioni elementari al giorno.			

Nel peggiore dei casi viene inserita una *Pianta* appartenente ad una *Specie* non ancora presente nel database.

Con ridondanza				
Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costrutto	Descrizione
1 * 2 = 2	Scrittura	E	Pianta	1 operazione di scrittura su <i>Pianta</i> per inserirne una nuova ennupla (inclusi <i>CodContenitore</i> e <i>CodTerreno</i>), collegandola alla relativa <i>Specie</i> .
1 * 2 = 2	Scrittura	E	Specie	1 operazione di scrittura su <i>Specie</i> per inserirne una nuova, se non già esistente. Viene incrementato l'attributo <i>NumeroPiante</i> .
4 * 2 = 8	Scrittura	R	ElementiNecessariSpecie	4 operazioni di scrittura su <i>ElementiNecessariSpecie</i> per associare alla specie gli elementi di cui ha bisogno.
2 * 2 = 4	Scrittura	R	PeriodiSpecie	2 operazioni di scrittura su <i>PeriodiSpecie</i> per associare alla <i>Specie</i> i periodi vegetativi e/ di riposo.
2 * 2 = 4	Scrittura	R	EsigenzaConcimazioneSpecie	2 operazioni di scrittura su <i>EsigenzaConcimazioneSpecie</i> per associare alla <i>Specie</i> le concimazioni che necessita.
20	Totale operazioni elementari per singola operazione.			
20 * 4 = 80	Totale operazioni elementari al giorno.			

Con l'introduzione di questa ridondanza, si passa da una media di $(8640000 + 948 + 72) = 8641020$ a $(10 + 1422 + 80) = 1512$ operazioni elementari giornaliere: risulta quindi conveniente mantenere tale ridondanza.

6.3 Ridondanza per l'operazione 3

Un attributo ridondante *PiantePresenti* nell'entità *Sezione* potrebbe migliorare l'efficienza della terza operazione. Infatti, la tavola degli accessi di tale operazione diventerebbe:

Senza ridondanza				
Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
1920	Lettura	E	Ripiano	Lettura di <i>CodSezione</i> di tutte le occorrenze dell'entità <i>Ripiano</i> , per trovare i ripiani appartenenti alla sezione di cui si desidera trovare il numero di piante appartenenti.
7680	Lettura	E	Contenitore	Lettura di <i>CodPianta</i> di tutte le occorrenze dell'entità <i>Contenitore</i> che abbiano come <i>CodRipiano</i> uno dei ripiani trovati precedentemente.
9600	Totale operazioni elementari per singola operazione.			
9600 * 15 = 144000	Totale operazioni elementari al giorno.			

Con ridondanza				
Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
1	Lettura	E	Sezione	Lettura di <i>PiantePresenti</i> dell'entità <i>Sezione</i> per ricavare il numero di piante presenti nella sezione.
1	Totale operazioni elementari per singola operazione.			
1 * 15 = 15	Totale operazioni elementari al giorno.			

Conviene notare che tale operazione potrebbe essere svolta per una *Serra* o una *Sede* (si suppone che non ci sia bisogno di una operazione del genere per un *Ripiano* in quanto potrebbe bastare un controllo visivo). Le tavole degli accessi per la terza operazione su queste due entità senza la ridondanza in questione richiede un numero di operazioni elementari maggiori in quanto *Serra* e *Sede* sono collegate da altre relazioni a partire da *Sezione*. Se l'introduzione della ridondanza risulterà conveniente per *Sezione*, quindi, lo sarà automaticamente anche per le entità *Serra* e *Sede*.

Per analizzare il costo di tale operazione, occorre valutare la tavola dei volumi relativa all'operazione 5, ovvero all'inserimento di una nuova pianta nel magazzino.

Come tavola dei volumi senza ridondanza verrà utilizzata quella descritta nel paragrafo precedente, ovvero quella con la ridondanza introdotta per l'operazione 2.

Come prima, il caso peggiore prevede che venga inserita una *Pianta* appartenente ad una *Specie* non ancora presente nel database.

Senza ridondanza				
Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione

1 * 2 = 2	Scrittura	E	Pianta	1 operazione di scrittura su <i>Pianta</i> per inserirne una nuova ennupla (inclusi <i>CodContentitore</i> e <i>CodTerreno</i>), collegandola alla relativa <i>Specie</i> .
1 * 2 = 2	Scrittura	E	Specie	1 operazione di scrittura su <i>Specie</i> per inserirne una nuova, se non già esistente. Viene incrementato l'attributo <i>NumeroPiante</i> .
4 * 2 = 8	Scrittura	R	ElementiNecessariSpecie	4 operazioni di scrittura su <i>ElementiNecessariSpecie</i> per associare alla specie gli elementi di cui ha bisogno.
2 * 2 = 4	Scrittura	R	PeriodiSpecie	2 operazioni di scrittura su <i>PeriodiSpecie</i> per associare alla <i>Specie</i> i periodi vegetativi e/ di riposo.
2 * 2 = 4	Scrittura	R	EsigenzaConcimazioneSpecie	2 operazioni di scrittura su <i>EsigenzaConcimazioneSpecie</i> per associare alla <i>Specie</i> le concimazioni che necessita.
20	Totale operazioni elementari per singola operazione.			
20 * 4 = 80	Totale operazioni elementari al giorno.			

Con l'inserimento della ridondanza, occorre solo aggiornare l'attributo *NumeroPiante* in questione:

Con ridondanza				
Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costrutto	Descrizione
1 * 2 = 2	Scrittura	E	Pianta	1 operazione di scrittura su <i>Pianta</i> per inserirne una nuova ennupla (inclusi <i>CodContentitore</i> e <i>CodTerreno</i>), collegandola alla relativa <i>Specie</i> .
1 * 2 = 2	Scrittura	E	Specie	1 operazione di scrittura su <i>Specie</i> per inserirne una nuova, se non già esistente. Viene incrementato l'attributo <i>NumeroPiante</i> .
4 * 2 = 8	Scrittura	R	ElementiNecessariSpecie	4 operazioni di scrittura su <i>ElementiNecessariSpecie</i> per associare alla specie gli elementi di cui ha bisogno.
2 * 2 = 4	Scrittura	R	PeriodiSpecie	2 operazioni di scrittura su <i>PeriodiSpecie</i> per associare alla <i>Specie</i> i periodi vegetativi e/ di riposo.
2 * 2 = 4	Scrittura	R	EsigenzaConcimazioneSpecie	2 operazioni di scrittura su <i>EsigenzaConcimazioneSpecie</i> per associare alla <i>Specie</i> le concimazioni che necessita.
1	Lettura	E	Contentitore	Lettura di <i>CodRipiano</i> di <i>Contentitore</i> per trovare il ripiano in cui il contenitore è collocato.

1	Lettura	E	Ripiano	Lettura di <i>CodSezione</i> di <i>Ripiano</i> per trovare la sezione in cui il ripiano è collocato.
1* 2 = 2	Scrittura	E	Sezione	Aggiornamento dell'attributo <i>PiantePresenti</i> dell'entità <i>Sezione</i> .
24	Totale operazioni elementari per singola operazione.			
20 * 4 = 96	Totale operazioni elementari al giorno.			

Si passa quindi da una media di $(144000 + 80) = 144080$ a $(15 + 96) = 111$ operazioni elementari giornaliere, perciò risulta quindi conveniente introdurre tali ridondanze (come precedentemente già discusso, anche per le entità *Serra* e *Sede*).

6.4 Ridondanza per l'operazione 4

Grazie all'introduzione della ridondanza per l'operazione 2, tale operazione si può semplificare aggiungendo un attributo *NumPianteVendute* all'entità *Specie*.

Di seguito, le tavole degli accessi:

Senza ridondanza				
Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
27648	Lettura	E	Scheda	Lettura di <i>CodPianta</i> di tutte le istanze di <i>Scheda</i> .
27648	Lettura	E	Pianta	Lettura di <i>Nome</i> di tutte le istanze di <i>Pianta</i> collegate alle istanze di <i>Scheda</i> .
55296	Totale operazioni elementari per singola operazione.			
55296 * 20 = 1105920	Totale operazioni elementari al giorno.			

Con ridondanza				
Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
90	Lettura	E	Specie	Lettura di <i>NumPianteVendute</i> di tutte le istanze di <i>Specie</i> .
90	Totale operazioni elementari per singola operazione.			
90 * 20 = 1800	Totale operazioni elementari al giorno.			

Per calcolare il costo della ridondanza, occorre valutare la tavola degli accessi per l'operazione di vendita di una *Pianta*. Dalla tavola dei volumi risulta che in 30 anni di attività dell'impresa sono state create 27648 schede, relativi a 27648 acquisti, ovvero $27648 / 30 = 921,6$ acquisti all'anno, ovvero $921,6 / 365 = 2,52$ acquisti ogni giorno. Si arrotonda a 3 acquisti al giorno per un maggior margine di sicurezza.

Nel caso senza ridondanza, nel caso peggiore la *Pianta* appena acquistata si troverà in un nuovo *Vaso*.

Senza ridondanza				
Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
1 * 2 = 2	Scrittura	E	Scheda	Creazione di una nuova istanza di <i>Scheda</i> relativa alla <i>Pianta</i> venduta.
1 * 2 = 2	Scrittura	E	Pianta	Scrittura di <i>CodScheda</i> della <i>Scheda</i> appena creata nella relativa istanza di <i>Pianta</i> .
1 * 2 = 2	Scrittura	E	Vaso	Creazione di una nuova istanza di <i>Vaso</i> nel quale la <i>Pianta</i> relativa alla <i>Scheda</i> è contenuta.
6	Totale operazioni elementari per singola operazione.			
6 * 3 = 18	Totale operazioni elementari al giorno.			

Con ridondanza				
Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
1 * 2 = 2	Scrittura	E	Scheda	Creazione di una nuova istanza di <i>Scheda</i> relativa alla <i>Pianta</i> venduta.
1 * 2 = 2	Scrittura	E	Pianta	Scrittura di <i>CodScheda</i> della <i>Scheda</i> appena creata nella relativa istanza di <i>Pianta</i> .
1 * 2 = 2	Scrittura	E	Vaso	Creazione di una nuova istanza di <i>Vaso</i> nel quale la <i>Pianta</i> relativa alla <i>Scheda</i> è contenuta.
1	Lettura	E	Pianta	Lettura del nome della specie dell'entità <i>Pianta</i> per trovarne la relativa <i>Specie</i> .
1 * 2 = 2	Scrittura	E	Specie	Aggiornamento dell'attributo <i>NumPianteVendute</i> dell'entità <i>Specie</i> .
9	Totale operazioni elementari per singola operazione.			
9 * 3 = 27	Totale operazioni elementari al giorno.			

Si ha un abbassamento in media delle operazioni elementari giornaliere da $(1105920 + 18) = 1105938$ a $(1800 + 27) = 1827$. Conviene perciò introdurre tale ridondanza.

6.5 Ridondanza per l'operazione 5

L'operazione non richiede un grande numero di operazioni elementari, perciò è stato scelto di non inserire nessuna ridondanza.

6.6 Ridondanza per l'operazione 6

L'operazione non richiede un grande numero di operazioni elementari, perciò è stato scelto di non inserire nessuna ridondanza.

6.7 Ridondanza per l'operazione 7

Un attributo ridondante *NumMedia* nell'entità *Thread* potrebbe migliorare l'efficienza della terza operazione. Infatti, la tavola degli accessi di tale operazione diventerebbe:

Senza ridondanza				
Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
6912	Lettura	R	MediaPost	Lettura di tutte le istanze della relazione <i>MediaPost</i> per trovare i post relativi ai media pubblicati.
6912	Lettura	E	Post	Lettura dell'attributo <i>CodThread</i> di <i>Post</i> per apprendere in che <i>Thread</i> il <i>Post</i> in questione appartiene.
13824	Totale operazioni elementari per singola operazione.			
13824 * 5 = 69120	Totale operazioni elementari al giorno.			

Con ridondanza				
Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
1	Lettura	E	Thread	Lettura di <i>NumMedia</i> dell'entità <i>Thread</i> per ricavarne il numero di contenuti multimediali contenuti.
1	Totale operazioni elementari per singola operazione.			
1 * 5 = 5	Totale operazioni elementari al giorno.			

Per calcolare il costo di aggiornamento di tale ridondanza, è necessario valutare la tavola degli accessi relativa all'inserimento di un post di un utente con contenuto multimediale.

Per la tavola degli accessi senza ridondanza si fa riferimento a quella analizzata nel paragrafo 5.3 *“Considerazione sulle ridondanze già presenti”* in presenza di ridondanza n° 4 (perché mantenuta), dove la medesima operazione viene analizzata per calcolare il costo della quarta ridondanza già esistente, dichiarata nel paragrafo 3.1.2 *“Attributi derivabili da attributi di un'altra entità”*.

Nel caso peggiore, il contenuto multimediale è nuovo. Per evidenziare il costo introdotto da tale ridondanza, si considera il caso in cui il *Thread* è già esistente (se non lo fosse, non si avrebbero costi di aggiornamento della ridondanza in quanto è in ogni caso necessario scrivere sull'entità *Thread*).

Senza ridondanza				
Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
1 * 2 = 2	Scrittura	E	Post	Creazione di una nuova istanza di <i>Post</i> .
1 * 2 = 2	Scrittura	E	Media	Creazione di una nuova istanza di <i>Media</i> .
1 * 2 = 2	Scrittura	R	MediaPost	Associazione tra il <i>Post</i> appena creato e il contenuto multimediale in esso contenuto.
1 * 2 = 2	Scrittura	E	Account	Aggiornamento di <i>NumPostPubblicati</i> nella relativa istanza di <i>Account</i> dell'utente che ha pubblicato il nuovo post.
8	Totale operazioni elementari per singola operazione.			
8 * 4 = 32	Totale operazioni elementari al giorno.			

Con ridondanza				
Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
1 * 2 = 2	Scrittura	E	Post	Creazione di una nuova istanza di <i>Post</i> .
1 * 2 = 2	Scrittura	E	Media	Creazione di una nuova istanza di <i>Media</i> .
1 * 2 = 2	Scrittura	R	MediaPost	Associazione tra il <i>Post</i> appena creato e il contenuto multimediale in esso contenuto.
1 * 2 = 2	Scrittura	E	Account	Aggiornamento della ridondanza <i>NumPostPubblicati</i> nella relativa istanza di <i>Account</i> dell'utente che ha pubblicato il nuovo post.
1 * 2 = 2	Scrittura	E	Thread	Aggiornamento della ridondanza <i>NumMedia</i> di <i>Thread</i> .
10	Totale operazioni elementari per singola operazione.			
10 * 4 = 40	Totale operazioni elementari al giorno.			

Tale ridondanza riduce il numero di operazioni elementari in media da $69120 + 32 = 69152$ a $5 + 40 = 45$.
 Conviene perciò introdurla.

6.8 Ridondanza per l'operazione 8

Per semplificare tale operazione si potrebbe aggiungere un attributo ridondante *NumeroEsordi* nell'entità *Specie* per mantenere il conto di tutte le volte che una *Pianta* appartenente ad una *Specie* si ammala.
 La tavola degli accessi dell'operazione diventerebbe:

Senza ridondanza				
Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
103680	Lettura	E	ReportDiagnostica	Lettura di <i>CodPianta</i> tutte le istanze di <i>ReportDiagnostica</i> per determinare la <i>Pianta</i> che più spesso si ammala.
103680	Lettura	E	Pianta	Lettura di <i>Nome</i> di <i>Pianta</i> per tutte le istanze presenti della relazione <i>ReportDiagnostica</i> .
207360	Totale operazioni elementari per singola operazione.			
207360 * 10 = 2073600	Totale operazioni elementari al giorno.			

Con ridondanza				
Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
90	Lettura	E	Specie	Lettura di <i>NumeroEsordi</i> di tutte le istanze dell'entità <i>Specie</i>
90	Totale operazioni elementari per singola operazione.			
90 * 10 = 900	Totale operazioni elementari al giorno.			

Per calcolare il costo introdotto dall'aggiornamento di tale ridondanza, occorre analizzare la tavola degli accessi relative all'operazione di aggiunta di un report di diagnostica di una pianta.

Osservando la tavola dei volumi si nota come in 30 anni di attività dell'impresa siano stati creati 103680

report di diagnostica, equivalenti a $103680 / 30 = 3456$ report annuali, ovvero $3456 / 365 = 9,47$ report giornalieri, approssimato a 10 report al giorno per un maggior margine di sicurezza.

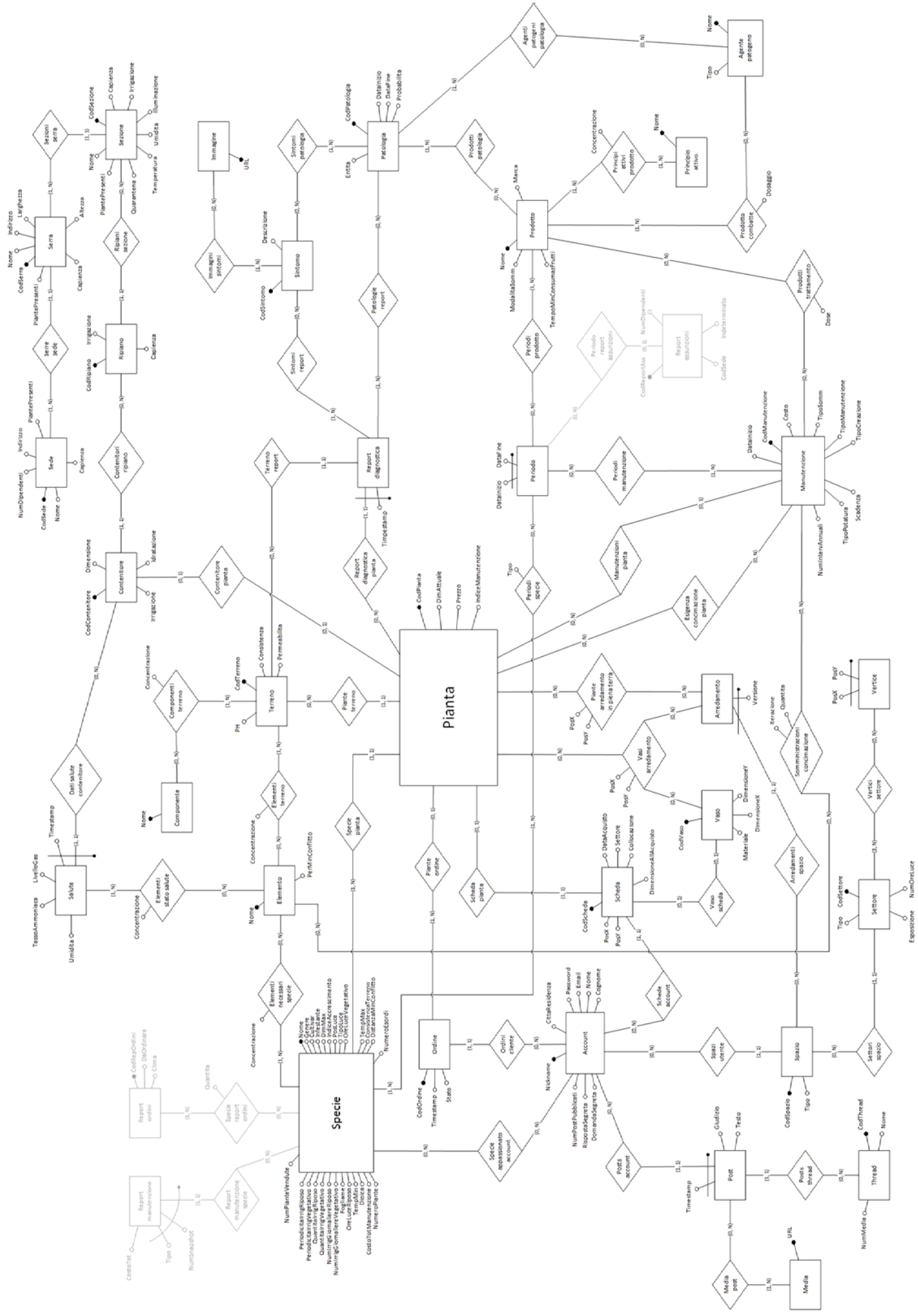
Senza ridondanza				
Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
1 * 2 = 2	Scrittura	E	ReportDiagnostica	Creazione di una nuova istanza dell'entità <i>ReportDiagnostica</i> .
2 * 2 = 4	Scrittura	R	SintomiReport	Associazione in media di due sintomi al report appena creato.
1,2 * 2 = 2,4	Scrittura	R	PatologieReport	Associazione di una diagnosi che prevede in media 1,2 patologie associate.
8,4	Totale operazioni elementari per singola operazione.			
8,4 * 10 = 84	Totale operazioni elementari al giorno.			

Con ridondanza				
Num. Operazioni Elementari	Tipo Operazione	E/R	Nome del costruito	Descrizione
1 * 2 = 2	Scrittura	E	ReportDiagnostica	Creazione di una nuova istanza dell'entità <i>ReportDiagnostica</i> .
2 * 2 = 4	Scrittura	R	SintomiReport	Associazione in media di due sintomi al report appena creato.
1,2 * 2 = 2,4	Scrittura	R	PatologieReport	Associazione di una diagnosi che prevede in media 1,2 patologie associate.
1	Lettura	E	Pianta	Lettura del nome della specie dell'entità <i>Pianta</i> .
1 * 2 = 2	Scrittura	E	Specie	Aggiornamento dell'attributo ridondante <i>NumeroEsordi</i> dell'entità <i>Specie</i> .
11,4	Totale operazioni elementari per singola operazione.			
11,4 * 10 = 114	Totale operazioni elementari al giorno.			

L'introduzione di tale ridondanza vede un calo di operazioni elementari giornaliere da una media di $2073600 + 84 = 2073684$ a $900 + 114 = 1014$, perciò conviene introdurre tale ridondanza.

6.9 Versione finale del database

Di seguito è riportata la versione finale del database, con tutte le ridondanze introdotte e discusse fino ad ora:



7. Progettazione logica

7.1 Traduzione verso il modello logico relazionale

Di seguito sono riportate le tabelle derivanti dallo schema E-R della versione finale del database. Vengono giustificati i soli accorpamenti (o non accorpamenti) che prevedono alternative.

- Pianta (CodPianta, DimAttuale, Prezzo, IndiceManutenzione, NomeSpecie, CodTerreno, CodContenitore, CodOrdine)
 - *L'associazione ContenitorePianta è stata accorpata in Pianta in quanto viene semplificata la quinta operazione, ovvero l'inserimento di una nuova pianta nel magazzino, e non è spesso di interesse ricavare quale Pianta è contenuta in un dato Contenitore.*
*È utile però poter sapere se un contenitore è vuoto o occupato: **viene per questo introdotto l'attributo Pieno nell'entità Contenitore.***
 - *Non accorpate l'associazione PiantaOrdine in Pianta non porta a nessun vantaggio.*
- Scheda (CodScheda, DataAcquisto, Settore, Collocazione, DimensioneAllAcquisto, PosX, PosY, CodPianta, CodVaso, Utente)
 - *L'associazione VasoScheda è stata accorpata in Scheda in quanto è una informazione strettamente collegata ad un Scheda, e non ha a che fare con il database dei Vasi.*
- Account (Nickname, CittaResidenza, Password, Email, Nome, Cognome, DomandaSegreta, RispostaSegreta, NumPostPubblicati)
- Ordine (CodOrdine, Timpestamp, Stato, Utente)
- Post (Timestamp, Utente, Giudizio, Testo, CodThread)
- Thread (CodThread, Nome, NumMedia)
- MediaPost (Timestamp, Utente, URL)
- Media (URL)
- SpecieAppassionatoAccount (NomeSpecie, Utente)
- Specie (Nome, Genere, Cultivar, Infestante, DimMax, IndiceAccrescimento, PosLuce, TipoLuce, OreLuceVegetativo, TempMax, ConsistenzaTerreno, DistanzaMinConflitto, NumeroEsordi, NumeroPiante, CostoTotManutenzione, Dioica, TempMin, OreLuceRiposo, Fogliame, NumIrrigGiornaliereVegetativo, NumIrrigGiornaliereRiposo, QuantalIrrigVegetativo, QuantalIrrigRiposo, PeriodicalIrrigVegetativo, PeriodicalIrrigRiposo, NumPianteVendute)
- ReportManutenzione (Tipo, NumSnapshot, NomeSpecie, CostTot)
- SpecieReportOrdini (CodRepOrdini, NomeSpecie, Quantita)
- ReportOrdini (CodRepOrdini, DaOrdinare, Clima)
- ElementiNecessariSpecie (NomeSpecie, NomeElemento, Concentrazione)
- Elemento (Nome, PerMinConflitto)
- ElementiTerreno (NomeElemento, CodTerreno, Concentrazione)
- Terreno (CodTerreno, PH, Consistenza, Permeabilita)
- ComponentiTerreno (CodTerreno, NomeComponente, Concentrazione)
- Componente (Nome)
- ElementiStatoSalute (CodContenitore, Timpestamp, NomeElemento, Concentrazione)
- Salute (Timestamp, CodContenitore, Umidita, TassoAmmoniacca, LivelloGas)
- Contenitore (CodContenitore, Dimensione, Idratazione, Irrigazione, CodRipiano, Pieno)
 - *L'attributo Pieno è nato dall'accorpamento dell'associazione ContenitorePianta nell'entità Pianta. Non interessa conoscere la Pianta contenuta in un dato Contenitore, ma solo se quest'ultimo è occupato o no.*
- Ripiano (CodRipiano, Capienza, Irrigazione, CodSezione)

- Sezione (CodSezione, Capienza, Irrigazione, Illuminazione, Umidita, Temperatura, Quarantena, PiantePresenti, Nome, CodSerra)
- Serra (CodSerra, Nome, Indirizzo, Larghezza, Altezza, Capienza, PiantePresenti, CodSede)
- Sede (CodSede, NumDipendenti, Indirizzo, PiantePresenti, Nome)
- Spazio (CodSpazio, Tipo, Utente)
- Settore (CodSettore, Tipo, Esposizione, NumOreLuce, CodSpazio)
- VerticiSettore (CodSettore, PosX, PosY)
- Vertice (PosX, PosY)
- Arredamento (Versione, CodSpazio)
- Vaso (CodVaso, Materiale, DimensioneX, DimensioneY)
- VasiArredamento (CodVaso, CodSpazio, Versione, CodPianta, PosX, PosY)
- PianteArredamentoInPienaTerra (CodPianta, CodSpazio, Versione, PosX, PosY)
- SomministrazioneConcimazione (NomeElemento, CodManutenzione, Iterazione, Quantita)
- EsigenzaConcimazionePianta (CodPianta, CodManutenzione)
- PeriodiSpecie (NomeSpecie, DataInizio, DataFine, Tipo)
- ReportDiagnostica (CodPianta, Timestamp, CodTerreno)
- SintomiReport (CodPianta, Timestamp, CodSintomo)
- PatologieReport (CodPianta, Timestamp, CodPatologia)
- ~~Periodo (DataInizio, DataFine)~~
 - *Questa relazione aveva come scopo solo quello di rappresentare, nello schema ER, 4 attributi multivalore.*
- PeriodiProdotti (NomeProdotto, DataInizio, DataFine)
- Prodotto (Nome, Marca, TempoMinConsumazFrutti, ModalitaSomm)
- PrincipiAttiviProdotto (NomeProdotto, NomePrincipioAttivo, Concentrazione)
- PrincipioAttivo (Nome)
- ProdottoCombatte (NomeProdotto, NomeAgentePatogeno, Dosaggio)
- AgentePatogeno (Nome, Tipo)
- AgentiPatogeniPatologia (CodPatologia, NomeAgentePatogeno)
- Patologia (CodPatologia, DataInizio, DataFine, Probabilita, Entita)
- ProdottiPatologia (CodPatologia, NomeProdotto)
- SintomiPatologia (CodPatologia, CodSintomo)
- Sintomo (CodSintomo, Descrizione)
- ImmaginiSintomi (CodSintomo, URL)
- Immagine (URL)
- ProdottiTrattamento (NomeProdotto, CodManutenzione, Dose)
- Manutenzione (CodManutenzione, DataInizio, Costo, TipoSomm, TipoManutenzione, TipoCreazione, Scadenza, TipoPotatura, NumIntervAnnuali, CodPianta)
 - *Non accorpate la relazione ManutenzioniPianta nella tabella Manutenzione non porta a nessun vantaggio.*
- PeriodiManutenzione (CodManutenzione, DataInizio, DataFine)
- ReportAssunzioni (CodReportAss, CodSede, NumDipendenti, Indeterminato, DataInizio, DataFine)
 - *Non accorpate la relazione PeriodoReportAssunzioni nella tabella ReportAssunzioni non porta a nessun vantaggio.*

7.2 Vincoli di integrità referenziale

- Esiste un vincolo di integrità referenziale tra l'attributo *NomeSpecie* della tabella *Pianta* e l'attributo *Nome* della tabella *Specie*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodTerreno* della tabella *Pianta* e l'attributo *CodTerreno* della tabella *Terreno*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodOrdine* della tabella *Pianta* e l'attributo *CodOrdine* della tabella *Ordine*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodContenitore* della tabella *Pianta* e l'attributo *CodContenitore* della tabella *Contenitore*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodPianta* della tabella *Scheda* e l'attributo *CodPianta* della tabella *Pianta*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodVaso* della tabella *Scheda* e l'attributo *CodVaso* della tabella *Vaso*.
- Esiste un vincolo di integrità referenziale tra l'attributo *Utente* della tabella *Scheda* e l'attributo *Nickname* della tabella *Account*.
- Esiste un vincolo di integrità referenziale tra l'attributo *Utente* della tabella *Ordine* e l'attributo *Nickname* della tabella *Account*.
- Esiste un vincolo di integrità referenziale tra l'attributo *Utente* della tabella *Post* e l'attributo *Nickname* della tabella *Account*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodThread* della tabella *Post* e l'attributo *CodThread* della tabella *Thread*.
- Esiste un vincolo di integrità referenziale tra la coppia di attributi *Timestamp* e *Utente* della tabella *MediaPost* e la coppia di attributi *Timestamp* e *Utente* della tabella *Post*.
- Esiste un vincolo di integrità referenziale tra l'attributo *URL* della tabella *MediaPost* e l'attributo *URL* della tabella *Media*.
- Esiste un vincolo di integrità referenziale tra l'attributo *NomeSpecie* della tabella *SpecieAppassionatoAccount* e l'attributo *Nome* della tabella *Specie*.
- Esiste un vincolo di integrità referenziale tra l'attributo *Utente* della tabella *SpecieAppassionatoAccount* e l'attributo *Nickname* della tabella *Account*.
- Esiste un vincolo di integrità referenziale tra l'attributo *NomeSpecie* della tabella *ReportManutenzione* e l'attributo *Nome* della tabella *Specie*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodRepOrdini* della tabella *SpecieReportOrdini* e l'attributo *CodRepOrdini* della tabella *ReportOrdini*.
- Esiste un vincolo di integrità referenziale tra l'attributo *NomeSpecie* della tabella *SpecieReportOrdini* e l'attributo *Nome* della tabella *Specie*.
- Esiste un vincolo di integrità referenziale tra l'attributo *NomeSpecie* della tabella *ElementiNecessariSpecie* e l'attributo *Nome* della tabella *Specie*.
- Esiste un vincolo di integrità referenziale tra l'attributo *NomeElemento* della tabella *ElementiNecessariSpecie* e l'attributo *Nome* della tabella *Elemento*.
- Esiste un vincolo di integrità referenziale tra l'attributo *NomeElemento* della tabella *ElementiTerreno* e l'attributo *Nome* della tabella *Elemento*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodTerreno* della tabella *ElementiTerreno* e l'attributo *CodTerreno* della tabella *Terreno*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodTerreno* della tabella *ComponentiTerreno* e l'attributo *CodTerreno* della tabella *Terreno*.

- Esiste un vincolo di integrità referenziale tra l'attributo *NomeComponente* della tabella *ComponentiTerreno* e l'attributo *Nome* della tabella *Componente*.
- Esiste un vincolo di integrità referenziale tra la coppia di attributi *CodContenitore* e *Timestamp* della tabella *ElementiStatoSalute* e la coppia di attributi *CodContenitore* della tabella *Contenitore* e *Timestamp* della tabella *Salute*.
- Esiste un vincolo di integrità referenziale tra l'attributo *NomeElemento* della tabella *ElementiStatoSalute* e l'attributo *Nome* della tabella *Elemento*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodContenitore* della tabella *Salute* e l'attributo *CodContenitore* della tabella *Contenitore*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodRipiano* della tabella *Contenitore* e l'attributo *CodRipiano* della tabella *Ripiano*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodSezione* della tabella *Ripiano* e l'attributo *CodSezione* della tabella *Sezione*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodSerra* della tabella *Sezione* e l'attributo *CodSerra* della tabella *Serra*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodSede* della tabella *Serra* e l'attributo *CodSede* della tabella *Sede*.
- Esiste un vincolo di integrità referenziale tra l'attributo *Utente* della tabella *Spazio* e l'attributo *Nickname* della tabella *Account*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodSettore* della tabella *VerticiSettore* e l'attributo *CodSettore* della tabella *Settore*.
- Esiste un vincolo di integrità referenziale tra la coppia di attributi *PosX* e *PosY* della tabella *VerticiSettore* e la coppia di attributi *PosX* e *PosY* della tabella *Vertice*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodSpazio* della tabella *Arredamento* e l'attributo *CodSpazio* della tabella *Spazio*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodVaso* della tabella *VasiArredamento* e l'attributo *CodVaso* della tabella *Vaso*.
- Esiste un vincolo di integrità referenziale tra la coppia di attributi *CodSpazio* e *Versione* della tabella *VasiArredamento* e tra la coppia di attributi *CodSpazio* e *Versione* della tabella *Arredamento*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodPianta* della tabella *VasiArredamento* e l'attributo *CodPianta* della tabella *Pianta*.
- Esiste un vincolo di integrità referenziale tra la coppia di attributi *CodSpazio* e *Versione* della tabella *PianteArredamentoInPienaTerra* e la coppia di attributi *CodSpazio* della tabella *Spazio* e *Versione* della tabella *Arredamento*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodPianta* della tabella *PianteArredamentoInPienaTerra* e l'attributo *CodPianta* della tabella *Pianta*.
- Esiste un vincolo di integrità referenziale tra l'attributo *NomeElemento* della tabella *SomministrazioneConcimazione* e l'attributo *Nome* della tabella *Elemento*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodManutenzione* della tabella *SomministrazioneConcimazione* e l'attributo *CodManutenzione* della tabella *Manutenzione*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodPianta* della tabella *EsigenzaConcimazionePianta* e l'attributo *CodPianta* della tabella *Pianta*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodManutenzione* della tabella *EsigenzaConcimazionePianta* e l'attributo *CodManutenzione* della tabella *Manutenzione*.
- Esiste un vincolo di integrità referenziale tra l'attributo *NomeSpecie* della tabella *PeriodiSpecie* e l'attributo *Nome* della tabella *Specie*.

- Esiste un vincolo di integrità referenziale tra l'attributo *CodPianta* della tabella *ReportDiagnostica* e l'attributo *CodPianta* della tabella *Pianta*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodTerreno* della tabella *ReportDiagnostica* e l'attributo *CodTerreno* della tabella *Terreno*.
- Esiste un vincolo di integrità referenziale tra la coppia di attributi *CodPianta* e *Timestamp* della tabella *SintomiReport* e la coppia di attributi *CodPianta* della tabella *Pianta* e *Timestamp* della tabella *ReportDiagnostica*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodSintomo* della tabella *SintomiReport* e l'attributo *CodSintomo* della tabella *Sintomo*.
- Esiste un vincolo di integrità referenziale tra la coppia di attributi *CodPianta* e *Timestamp* della tabella *PatologieReport* e la coppia di attributi *CodPianta* e *Timestamp* della tabella *ReportDiagnostica*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodPatologia* della tabella *PatologieReport* e l'attributo *CodPatologia* della tabella *Patologia*.
- Esiste un vincolo di integrità referenziale tra l'attributo *NomeProdotto* della tabella *PeriodiProdotti* e l'attributo *Nome* della tabella *Prodotto*.
- Esiste un vincolo di integrità referenziale tra l'attributo *NomeProdotto* della tabella *PrincipiAttiviProdotto* e l'attributo *Nome* della tabella *Prodotto*.
- Esiste un vincolo di integrità referenziale tra l'attributo *NomePrincipioAttivo* della tabella *PrincipiAttiviProdotto* e l'attributo *Nome* della tabella *PrincipioAttivo*.
- Esiste un vincolo di integrità referenziale tra l'attributo *NomeProdotto* della tabella *ProdottoCombatte* e l'attributo *Nome* della tabella *Prodotto*.
- Esiste un vincolo di integrità referenziale tra l'attributo *NomeAgentePatogeno* della tabella *ProdottoCombatte* e l'attributo *Nome* della tabella *AgentePatogeno*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodPatologia* della tabella *AgentiPatogeniPatologia* e l'attributo *CodPatologia* della tabella *Patologia*.
- Esiste un vincolo di integrità referenziale tra l'attributo *NomeAgentePatogeno* della tabella *AgentiPatogeniPatologia* e l'attributo *Nome* della tabella *AgentePatogeno*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodPatologia* della tabella *ProdottiPatologia* e l'attributo *CodPatologia* della tabella *Patologia*.
- Esiste un vincolo di integrità referenziale tra l'attributo *NomeProdotto* della tabella *ProdottiPatologia* e l'attributo *Nome* della tabella *Prodotto*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodPatologia* della tabella *SintomiPatologia* e l'attributo *CodPatologia* della tabella *Patologia*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodSintomo* della tabella *SintomiPatologia* e l'attributo *CodSintomo* della tabella *Sintomo*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodSintomo* della tabella *ImmaginiSintomi* e l'attributo *CodSintomo* della tabella *Sintomo*.
- Esiste un vincolo di integrità referenziale tra l'attributo *URL* della tabella *ImmaginiSintomi* e l'attributo *URL* della tabella *Immagine*.
- Esiste un vincolo di integrità referenziale tra l'attributo *NomeProdotto* della tabella *ProdottiTrattamento* e l'attributo *Nome* della tabella *Prodotto*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodManutenzione* della tabella *ProdottiTrattamento* e l'attributo *CodManutenzione* della tabella *Manutenzione*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodPianta* della tabella *Manutenzione* e l'attributo *CodPianta* della tabella *Pianta*.

- Esiste un vincolo di integrità referenziale tra l'attributo *CodManutenzione* della tabella *PeriodiManutenzione* e l'attributo *CodManutenzione* della tabella *Manutenzione*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodSede* della tabella *ReportAssunzioni* e l'attributo *CodSede* della tabella *Sede*.
- Esiste un vincolo di integrità referenziale tra l'attributo *CodSpazio* della tabella *Settore* e l'attributo *CodSpazio* della tabella *Spazio*.

8. Analisi delle dipendenze funzionali e normalizzazione

Tabella Pianta:

Pianta (CodPianta, DimAttuale, Prezzo, IndiceManutenzione, NomeSpecie, CodTerreno, CodContenitore)

- CodPianta → DimAttuale, Prezzo, IndiceManutenzione, NomeSpecie, CodTerreno, CodContenitore

Poiché la parte sinistra è superchiave, *Pianta* è in BCNF.

Tabella Scheda:

Scheda (CodScheda, DataAcquisto, Settore, Collocazione, DimensioneAllAcquisto, PosX, PosY, CodPianta, CodVaso, Utente)

- CodScheda → DataAcquisto, Settore, Collocazione, DimensioneAllAcquisto, PosX, PosY, CodPianta, CodVaso, Utente

Poiché la parte sinistra è superchiave, *Scheda* è in BCNF.

Tabella Account:

Account (Nickname, CittaResidenza, Password, Email, Nome, Cognome, DomandaSegreta, RispostaSegreta, NumPostPubblicati)

- Nickname → CittaResidenza, Password, Email, Nome, Cognome, DomandaSegreta, RispostaSegreta, NumPostPubblicati

Poiché la parte sinistra è superchiave, *Account* è in BCNF.

Tabella Ordine:

Ordine (CodOrdine, Timestamp, Stato, Utente)

- CodOrdine → Timestamp, Stato, Utente

Poiché la parte sinistra è superchiave, *Ordine* è in BCNF.

Tabella Post:

Post (Timestamp, Utente, Giudizio, Testo, CodThread)

- Timestamp, Utente → Giudizio, Testo, CodThread

Poiché la parte sinistra è superchiave, *Post* è in BCNF.

Tabella Thread:

Thread (CodThread, Nome, NumMedia)

- CodThread → Nome, NumMedia

Poiché la parte sinistra è superchiave, *Thread* è in BCNF.

Tabella MediaPost:

MediaPost (Timestamp, Utente, URL)

Poiché non esistono dipendenze funzionali, *MediaPost* è in BCNF.

Tabella Media:

Media (URL)

Poiché non esistono dipendenze funzionali, *Media* è in BCNF.

Tabella SpecieAppassionatoAccount:

SpecieAppassionatoAccount (NomeSpecie, Utente)

Poiché non esistono dipendenze funzionali, *SpecieAppassionatoAccount* è in BCNF.

Tabella Specie:

Specie (Nome, Genere, Cultivar, Infestante, DimMax, IndiceAccrescimento, PosLuce, TipoLuce, OreLuceVegetativo, TempMax, ConsistenzaTerreno, DistanzaMinConflitto, NumeroEsordi, NumeroPiante, CostoTotManutenzione, Dioica, TempMin, OreLuceRiposo, Fogliame, NumIrrigGiornaliereVegetativo, NumIrrigGiornaliereRiposo, QuantalIrrigVegetativo, QuantalIrrigRiposo, PeriodicalIrrigVegetativo, PeriodicalIrrigRiposo, NumPianteVendute)

- Nome → Genere, Cultivar, Infestante, DimMax, IndiceAccrescimento, PosLuce, TipoLuce, OreLuceVegetativo, TempMax, ConsistenzaTerreno, DistanzaMinConflitto, NumeroEsordi, NumeroPiante, CostoTotManutenzione, Dioica, TempMin, OreLuceRiposo, Fogliame, NumIrrigGiornaliereVegetativo, NumIrrigGiornaliereRiposo, QuantalIrrigVegetativo, QuantalIrrigRiposo, PeriodicalIrrigVegetativo, PeriodicalIrrigRiposo, NumPianteVendute

Poiché la parte sinistra è superchiave, *Specie* è in BCNF.

Tabella ReportManutenzione:

ReportManutenzione (Tipo, NumSnapshot, NomeSpecie, CostoTot)

- Tipo, NumSnapshot, NomeSpecie → CostoTot

Poiché la parte sinistra è superchiave, *ReportManutenzione* è in BCNF.

Tabella SpecieReportOrdini:

SpecieReportOrdini (CodRepOrdini, NomeSpecie, Quantita)

- CodRepOrdini, NomeSpecie → Quantita

Poiché la parte sinistra è superchiave, *SpecieReportOrdini* è in BCNF.

Tabella ReportOrdini:

ReportOrdini (CordRepOrdini, DaOrdinare, Clima)

- CordRepOrdini → DaOrdinare, Clima

Poiché la parte sinistra è superchiave, *ReportOrdini* è in BCNF.

Tabella ElementiNecessariSpecie:

ElementiNecessariSpecie (NomeSpecie, NomeElemento, Concentrazione)

- NomeSpecie, NomeElemento → Concentrazione

Poiché la parte sinistra è superchiave, *ElementiNecessariSpecie* è in BCNF.

Tabella Elemento:

Elemento (Nome, PerMinConflitto)

- Nome → PerMinConflitto

Poiché la parte sinistra è superchiave, *Elemento* è in BCNF.

Tabella ElementiTerreno:

ElementiTerreno (NomeElemento, CodTerreno, Concentrazione)

- NomeElemento, CodTerreno → Concentrazione

Poiché la parte sinistra è superchiave, *ElementiTerreno* è in BCNF.

Tabella Terreno:

Terreno (CodTerreno, PH, Consistenza, Permeabilita)

- CodTerreno → PH, Consistenza, Permeabilita

Poiché la parte sinistra è superchiave, *Terreno* è in BCNF.

Tabella ComponentiTerreno:

ComponentiTerreno (CodTerreno, NomeComponente, Concentrazione)

- CodTerreno, NomeComponente → Concentrazione

Poiché la parte sinistra è superchiave, *ComponentiTerreno* è in BCNF.

Tabella Componente:

Componente (Nome)

Poiché non esistono dipendenze funzionali, *Componente* è in BCNF.

Tabella ElementiStatoSalute:

ElementiStatoSalute (CodContenitore, Timpestamp, NomeElemento, Concentrazione)

- CodContenitore, Timpestamp, NomeElemento → Concentrazione

Poiché la parte sinistra è superchiave, *ElementiStatoSalute* è in BCNF.

Tabella Salute:

Salute (Timestamp, CodContenitore, Umidita, TassoAmmoniaca, LivelloGas)

- Timestamp, CodContenitore → Umidita, TassoAmmoniaca, LivelloGas

Poiché la parte sinistra è superchiave, *Salute* è in BCNF.

Tabella Contenitore:

Contenitore (CodContenitore, Dimensione, Idratazione, Irrigazione, CodRipiano, Pieno)

- CodContenitore → Dimensione, Idratazione, Irrigazione, CodRipiano, Pieno

Poiché la parte sinistra è superchiave, *Contenitore* è in BCNF.

Tabella Ripiano:

Ripiano (CodRipiano, Capienza, Irrigazione, CodSezione)

- CodRipiano → Irrigazione, CodSezione

Poiché la parte sinistra è superchiave, *Ripiano* è in BCNF.

Tabella Sezione:

Sezione (CodSezione, Capienza, Irrigazione, Illuminazione, Umidita, Temperatura, Quarantena, PiantePresenti, Nome, CodSerra)

- CodSezione → Capienza, Irrigazione, Illuminazione, Umidità, Temperatura, Quarantena, PiantePresenti, Nome, CodSerra

Poiché la parte sinistra è superchiave, *Sezione* è in BCNF.

Tabella Serra:

Serra (CodSerra, Nome, Indirizzo, Larghezza, Altezza, Capienza, PiantePresenti, CodSede)

- CodSerra → Nome, Indirizzo, Larghezza, Altezza, Capienza, PiantePresenti, CodSede

Poiché la parte sinistra è superchiave, *Serra* è in BCNF.

Tabella Sede:

Sede (CodSede, Capienza, NumDipendenti, Indirizzo, PiantePresenti, Nome)

- CodSede → Capienza, NumDipendenti, Indirizzo, PiantePresenti, Nome

Poiché la parte sinistra è superchiave, *Sede* è in BCNF.

Tabella Spazio:

Spazio (CodSpazio, Tipo, Utente)

- CodSpazio → Tipo, Utente

Poiché la parte sinistra è superchiave, *Spazio* è in BCNF.

Tabella Settore:

Settore (CodSettore, Tipo, Esposizione, NumOreLuce, CodSpazio)

- CodSettore → Tipo, Esposizione, NumOreLuce, CodSpazio

Poiché la parte sinistra è superchiave, *Settore* è in BCNF.

Tabella VerticiSettore:

VerticiSettore (CodSettore, PosX, PosY)

Poiché non esistono dipendenze funzionali, *VerticiSettore* è in BCNF.

Tabella Vertice:

Vertice (PosX, PosY)

Poiché non esistono dipendenze funzionali, *Vertice* è in BCNF.

Tabella Arredamento:

Arredamento (Versione, CodSpazio)

Poiché non esistono dipendenze funzionali, *Arredamento* è in BCNF.

Tabella Vaso:

Vaso (CodVaso, Materiale, DimensioneX, DimensioneY)

- CodVaso → Materiale, DimensioneX, DimensioneY

Poiché la parte sinistra è superchiave, *Vaso* è in BCNF.

Tabella VasiArredamento:

VasiArredamento (CodVaso, CodSpazio, Versione, CodPianta, PosX, PosY)

- CodVaso, CodSpazio, Versione, CodPianta → PosX, PosY

Poiché la parte sinistra è superchiave, *VasiArredamento* è in BCNF.

Tabella PiantaArredamentoInPienaTerra:

PiantaArredamentoInPienaTerra (CodPianta, CodSpazio, Versione, PosX, PosY)

- CodPianta, CodSpazio, Versione → PosX, PosY

Poiché la parte sinistra è superchiave, *PiantaArredamentoInPienaTerra* è in BCNF.

Tabella SomministrazioneConcimazione:

SomministrazioneConcimazione (NomeElemento, CodManutenzione, Iterazione, Quantita)

- NomeElemento, CodManutenzione, Iterazione → Quantita

Poiché la parte sinistra è superchiave, *SomministrazioneConcimazione* è in BCNF.

Tabella EsigenzaConcimazionePianta:

EsigenzaConcimazionePianta (CodPianta, CodManutenzione)

Poiché non esistono dipendenze funzionali, *EsigenzaConcimazionePianta* è in BCNF.

Tabella PeriodiSpecie:

PeriodiSpecie (NomeSpecie, DataInizio, DataFine, Tipo)

- NomeSpecie, DataInizio, DataFine → Tipo

Poiché la parte sinistra è superchiave, *PeriodiSpecie* è in BCNF.

Tabella ReportDiagnostica:

ReportDiagnostica (CodPianta, Timestamp, CodTerreno)

- CodPianta, Timestamp → CodTerreno

Poiché la parte sinistra è superchiave, *ReportDiagnostica* è in BCNF.

Tabella SintomiReport:

SintomiReport (CodPianta, Timestamp, CodSintomo)

Poiché non esistono dipendenze funzionali, *SintomiReport* è in BCNF.

Tabella PatologieReport:

PatologieReport (CodPianta, Timestamp, CodPatologia)

Poiché non esistono dipendenze funzionali, *PatologieReport* è in BCNF.

Tabella PeriodiProdotti:

PeriodiProdotti (NomeProdotto, DataInizio, DataFine)

Poiché non esistono dipendenze funzionali, *PeriodiProdotti* è in BCNF.

Tabella Prodotto:

Prodotto (Nome, Marca, TempoMinConsumazFrutti, ModalitaSomm)

- Nome → Marca, TempoMinConsumazFrutti, ModalitaSomm

Poiché la parte sinistra è superchiave, *Prodotto* è in BCNF.

Tabella PrincipiAttiviProdotto:

PrincipiAttiviProdotto (NomeProdotto, NomePrincipioAttivo, Concentrazione)

- NomeProdotto, NomePrincipioAttivo → Concentrazione

Poiché la parte sinistra è superchiave, *PrincipiAttiviProdotto* è in BCNF.

Tabella PrincipioAttivo:

PrincipioAttivo (Nome)

Poiché non esistono dipendenze funzionali, *PrincipioAttivo* è in BCNF.

Tabella ProdottoCombatte:

ProdottoCombatte (NomeProdotto, NomeAgentePatogeno, Dosaggio)

- NomeProdotto, NomeAgentePatogeno → Dosaggio

Poiché la parte sinistra è superchiave, *ProdottoCombatte* è in BCNF.

Tabella AgentePatogeno:

AgentePatogeno (Nome, Tipo)

- Nome → Tipo

Poiché la parte sinistra è superchiave, *AgentePatogeno* è in BCNF.

Tabella AgentiPatogeniPatologia:

AgentiPatogeniPatologia (CodPatologia, NomeAgentePatogeno)

Poiché non esistono dipendenze funzionali, *AgentiPatogeniPatologia* è in BCNF.

Tabella Patologia:

Patologia (CodPatologia, DataInizio, DataFine, Probabilita, Entita)

- CodPatologia → DataInizio, DataFine, Probabilita, Entita

Poiché la parte sinistra è superchiave, *Patologia* è in BCNF.

Tabella ProdottiPatologia:

ProdottiPatologia (CodPatologia, NomeProdotto)

Poiché non esistono dipendenze funzionali, *ProdottiPatologia* è in BCNF.

Tabella SintomiPatologia:

SintomiPatologia (CodPatologia, CodSintomo)

Poiché non esistono dipendenze funzionali, *SintomiPatologia* è in BCNF.

Tabella Sintomo:

Sintomo (CodSintomo, Descrizione)

- CodSintomo → Descrizione

Poiché la parte sinistra è superchiave, *Sintomo* è in BCNF.

Tabella ImmaginiSintomi:

ImmaginiSintomi (CodSintomo, URL)

Poiché non esistono dipendenze funzionali, *ImmaginiSintomi* è in BCNF.

Tabella Immagine:

Immagine (URL)

Poiché non esistono dipendenze funzionali, *Immagine* è in BCNF.

Tabella ProdottiTrattamento:

ProdottiTrattamento (NomeProdotto, CodManutenzione, Dose)

- NomeProdotto, CodManutenzione → Dose

Poiché la parte sinistra è superchiave, *ProdottiTrattamento* è in BCNF.

Tabella Manutenzione:

Manutenzione (CodManutenzione, DataInizio, Costo, TipoSomm, TipoManutenzione, TipoCreazione, Scadenza, TipoPotatura, NumIntervAnnuali, CodPianta)

- CodManutenzione → DataInizio, Costo, TipoSomm, TipoManutenzione, TipoCreazione, Scadenza, TipoPotatura, NumIntervAnnuali, CodPianta

Poiché la parte sinistra è superchiave, *Manutenzione* è in BCNF.

Tabella PeriodiManutenzione:

PeriodiManutenzione (CodManutenzione, DataInizio, DataFine)

Poiché non esistono dipendenze funzionali, *PeriodiManutenzione* è in BCNF.

Tabella ReportAssunzioni:

ReportAssunzioni (CodReportAss, CodSede, NumDipendenti, Indeterminato, DataInizio, DataFine)

- CodReportAss → CodSede, NumDipendenti, Indeterminato, DataInizio, DataFine

Poiché la parte sinistra è superchiave, *ReportAssunzioni* è in BCNF.

9. Implementazione su DBMS Oracle MySQL

9.1 Business rules e vincoli di integrità generici

9.1.1 Definizione dell'indice di accrescimento di una specie

Con *indice di accrescimento* si intende la velocità con cui una specie raggiunge la sua dimensione massima, ovvero:

$$\text{IndiceAccrescimento} = \frac{\text{DimMax} \cdot 100}{\text{TempoImpiegato}}$$

Si può supporre che il tempo impiegato sia proporzionale alla quantità d'acqua impiegata per la sua irrigazione in un anno, dando meno importanza alla quantità ricevuta durante il periodo di riposo:

$$\begin{aligned} \text{TempoImpiegato} &= \frac{\text{NumIrrigGiornaliereVegetativo} \cdot \text{QuantitaIrrigVegetativo}}{365} \\ &+ \frac{1}{2} \left(\text{NumIrrigGiornaliereRiposo} \cdot \text{QuantitaIrrigRiposo} \cdot \frac{365}{\text{PeriodicitaIrrigRiposo}} \right) \end{aligned}$$

Effettuando alcune prove con dei valori tipi, il valore dell'indice di accrescimento è di solito compreso da 1 a 20.

9.1.2 Definizione dell'indice di manutenzione di una pianta

L'indice di manutenzione di una pianta è sicuramente direttamente proporzionale all'indice di accrescimento della specie a cui fa parte. È inversamente proporzionale invece alla grandezza che la pianta ha nel momento, rispetto alla dimensione massima che può raggiungere.

L'espressione che lega questi fattori all'indice di manutenzione sarà quindi:

$$IndiceManutenzione = \frac{IndiceAccrescimento}{(DimAttuale/DimMax)} = \frac{IndiceAccrescimento \cdot DimMax}{DimAttuale}$$

9.1.3 Limite sulla distanza minima di conflitto di una specie

La distanza minima di conflitto di una specie dev'essere maggiore o uguale alla metà della dimensione massima della pianta (quest'ultima indica il maggior diametro tra la parte radicale e aerea).

9.1.4 Controllo del range delle temperature accettabili da una specie

Ovviamente la temperatura minima dev'essere inferiore della temperatura massima.

Inoltre, la temperatura della sezione nella quale una pianta viene inserita, deve rientrare nel range di temperature accettabili dalle piante della stessa specie.

9.1.5 Controllo sulle password degli accounts

Le password devono almeno contenere 8 caratteri, come nella maggior parte dei siti web, e non possono essere uguali al Nickname scelte dall'utente, per motivi di sicurezza.

9.1.6 Controllo sulla risposta segreta dell'account

La risposta segreta scelta dall'utente non può essere uguale alla domanda segreta.

9.1.7 Controllo sulla percentuale degli elementi e delle componenti nel terreno

Le somme delle percentuali di elementi presenti in un terreno non può superare il 100%, così come quelle delle componenti che lo compongono.

9.1.8 Controllo sulla data di scadenza di una manutenzione

La scadenza di una manutenzione non può avvenire prima dell'inizio di quest'ultima.

9.1.9 Controllo sulle dimensioni del contenitore

Nel momento in cui una pianta viene inserita in un contenitore, occorre assicurarsi che l'attuale dimensione della pianta sia minore di quella del contenitore in cui andrà inserita.

9.1.10 Controllo sulla capienza dei ripiani, delle sezioni e delle serre

Occorre verificare che ci sia spazio per le piante inserite nelle serre, rispettando la capienza massima.

9.2 Script MySQL

```
SET NAMES latin1;
DROP DATABASE IF EXISTS `impresagiardinaggio`;
CREATE DATABASE IF NOT EXISTS `impresagiardinaggio`;
USE `impresagiardinaggio`;
SET FOREIGN_KEY_CHECKS = 1;
SET GLOBAL EVENT_SCHEDULER = ON;

--
-- Creazione tabella `account`
--
DROP TABLE IF EXISTS `account`;
CREATE TABLE `account` (
  `Nickname` char(50) NOT NULL,
  `CittaResidenza` char(50) DEFAULT NULL,
  `Password` char(50) NOT NULL,
  `Email` char(50) NOT NULL,
  `Nome` char(50) NOT NULL,
  `Cognome` char(50) NOT NULL,
  `DomandaSegreta` char(50) NOT NULL,
  `RispostaSegreta` char(50) NOT NULL,
  `NumPostPubblicati` int(11) unsigned NOT NULL DEFAULT '0',
  UNIQUE (`Email`),
  PRIMARY KEY (`Nickname`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Controllo password
DROP TRIGGER IF EXISTS `ControlloPassword`;
DELIMITER $$
CREATE TRIGGER `ControlloPassword` BEFORE INSERT ON `account` FOR EACH ROW
BEGIN
  IF(LENGTH(NEW.`Password`) < 8) THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Password troppo corta.';
  END IF;

  IF(NEW.`Nickname` = NEW.`Password`) THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT =
      'Impossibile scegliere la password uguale al nickname.';
  END IF;
END $$
DELIMITER ;

-- Controllo domanda e risposta segreta
DROP TRIGGER IF EXISTS `ControlloDomandaRispostaSegreta`;
DELIMITER $$
CREATE TRIGGER `ControlloDomandaRispostaSegreta`
  BEFORE INSERT ON `account` FOR EACH ROW
BEGIN
  IF(NEW.`DomandaSegreta` = NEW.`RispostaSegreta`) THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Impossibile scegliere la domanda segreta
      uguale alla risposta segreta.';
  END IF;
END $$
DELIMITER ;

--
-- Creazione tabella `agentepatogeno`
```

```

--
DROP TABLE IF EXISTS `agentepatogeno`;
CREATE TABLE `agentepatogeno` (
  `Nome` char(50) NOT NULL,
  `Tipo` char(50) NOT NULL,
  PRIMARY KEY (`Nome`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `agentipatogenipatologia`
--
DROP TABLE IF EXISTS `agentipatogenipatologia`;
CREATE TABLE `agentipatogenipatologia` (
  `CodPatologia` int(11) unsigned NOT NULL,
  `NomeAgentePatogeno` char(50) NOT NULL,
  PRIMARY KEY (`CodPatologia`,`NomeAgentePatogeno`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `arredamento`
--
DROP TABLE IF EXISTS `arredamento`;
CREATE TABLE `arredamento` (
  `Versione` int(11) unsigned NOT NULL,
  `CodSpazio` int(11) unsigned NOT NULL,
  PRIMARY KEY (`Versione`,`CodSpazio`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `componente`
--
DROP TABLE IF EXISTS `componente`;
CREATE TABLE `componente` (
  `Nome` char(50) NOT NULL,
  PRIMARY KEY (`Nome`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `componentiterreno`
--
DROP TABLE IF EXISTS `componentiterreno`;
CREATE TABLE `componentiterreno` (
  `CodTerreno` int(11) unsigned NOT NULL,
  `NomeComponente` char(50) NOT NULL,
  `Concentrazione` float(13,2) unsigned NOT NULL, -- mg/m^3
  PRIMARY KEY (`CodTerreno`,`NomeComponente`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Controllo sulle percentuali delle componenti di un terreno
DROP TRIGGER IF EXISTS `ControllaPercentualiComponenti`;
DELIMITER $$
CREATE TRIGGER `ControllaPercentualiComponenti`
  BEFORE INSERT ON `componentiterreno` FOR EACH ROW
BEGIN
  SET @PercentualeComponentiTerreno = (
    SELECT SUM(`Concentrazione`)
    FROM `componentiterreno`
    WHERE `CodTerreno` =
      NEW.`CodTerreno`);

  IF(@PercentualeComponentiTerreno + NEW.`Concentrazione` > 100) THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'La concentrazione di componenti in un terreno

```

```

        non puo\' superare il 100%.';
    END IF;
END $$
DELIMITER ;

--
-- Creazione tabella `contenitore`
--
DROP TABLE IF EXISTS `contenitore`;
CREATE TABLE `contenitore` (
  `CodContenitore` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `Dimensione` int(11) unsigned NOT NULL, -- centimetri (diametro)
  `Idratazione` float (13, 2) NOT NULL, -- percentuale
  `Irrigazione` float (13, 2) NOT NULL, -- percentuale
  `CodRipiano` int(11) unsigned DEFAULT NULL,
  `Pieno` tinyint(1) NOT NULL DEFAULT '0',
  PRIMARY KEY (`CodContenitore`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `elementinecessarispecie`
--
DROP TABLE IF EXISTS `elementinecessarispecie`;
CREATE TABLE `elementinecessarispecie` (
  `NomeSpecie` char(50) NOT NULL,
  `NomeElemento` char(50) NOT NULL,
  `Concentrazione` float(13, 2) unsigned NOT NULL, -- mg/m^3
  PRIMARY KEY (`NomeSpecie`,`NomeElemento`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `elementistatosalute`
--
DROP TABLE IF EXISTS `elementistatosalute`;
CREATE TABLE `elementistatosalute` (
  `CodContenitore` int(11) unsigned NOT NULL,
  `Timestamp` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
  ON UPDATE CURRENT_TIMESTAMP,
  `NomeElemento` char(50) NOT NULL,
  `Concentrazione` float(13, 2) unsigned NOT NULL, -- mg/m^3
  PRIMARY KEY (`CodContenitore`,`Timestamp`,`NomeElemento`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `elementiterreno`
--
DROP TABLE IF EXISTS `elementiterreno`;
CREATE TABLE `elementiterreno` (
  `NomeElemento` char(50) NOT NULL,
  `CodTerreno` int(11) unsigned NOT NULL,
  `Concentrazione` float(13, 2) unsigned NOT NULL, -- mg/m^3
  PRIMARY KEY (`NomeElemento`,`CodTerreno`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Controllo sulle percentuali degli elementi di un terreno
DROP TRIGGER IF EXISTS `ControllaPercentualiElementi`;
DELIMITER $$
CREATE TRIGGER `ControllaPercentualiElementi`
  BEFORE INSERT ON `elementiterreno` FOR EACH ROW
BEGIN
  SET @PercentualeElementiTerreno = ( SELECT SUM(`Concentrazione`)
    FROM `elementiterreno`

```

```

WHERE `CodTerreno` =
NEW.`CodTerreno`);

IF(@PercentualeElementiTerreno + NEW.`Concentrazione` > 100) THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'La concentrazione di elementi in un terreno
non puo\' superare il 100%.';
END IF;
END $$
DELIMITER ;

--
-- Creazione tabella `elemento`
--
DROP TABLE IF EXISTS `elemento`;
CREATE TABLE `elemento` (
    `Nome` char(50) NOT NULL,
    `PerMinConflitto` float(13, 2) unsigned NOT NULL,
    PRIMARY KEY (`Nome`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `esigenzaconcimazionepianta`
--
DROP TABLE IF EXISTS `esigenzaconcimazionepianta`;
CREATE TABLE `esigenzaconcimazionepianta` (
    `CodPianta` int(11) unsigned NOT NULL,
    `CodManutenzione` int(11) unsigned NOT NULL,
    PRIMARY KEY (`CodPianta`,`CodManutenzione`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `immagine`
--
DROP TABLE IF EXISTS `immagine`;
CREATE TABLE `immagine` (
    `URL` char(50) NOT NULL,
    PRIMARY KEY (`URL`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `immaginisintomi`
--
DROP TABLE IF EXISTS `immaginisintomi`;
CREATE TABLE `immaginisintomi` (
    `CodSintomo` int(11) unsigned NOT NULL,
    `URL` char(50) NOT NULL,
    PRIMARY KEY (`CodSintomo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `manutenzione`
--
DROP TABLE IF EXISTS `manutenzione`;
CREATE TABLE `manutenzione` (
    `CodManutenzione` int(11) unsigned NOT NULL AUTO_INCREMENT,
    `DataInizio` date NOT NULL,
    `Costo` int(11) NOT NULL, -- euro
    `TipoSomm` char(50) DEFAULT NULL, -- disciolto, nebulizzato
    `TipoManutenzione` char(50) NOT NULL, /* potatura, rinvaso,
concimazione, trattamento */
    `TipoCreazione` char(50) NOT NULL, /* su richiesta, programmata,

```

```

                                automatica */
`Scadenza` date DEFAULT NULL,
-- it.wikipedia.org/wiki/Potatura#Metodi_di_potatura
`TipoPotatura` char(50) DEFAULT NULL,
`NumIntervAnnuali` int(11) unsigned NOT NULL,
`CodPianta` int(11) unsigned DEFAULT NULL,
PRIMARY KEY (`CodManutenzione`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/* Aggiornamento della ridondanza `CostoTotManutenzione`
dell'entità `specie` */
DROP TRIGGER IF EXISTS `AggiornamentoCostoTotManutenzione`;
DELIMITER $$
CREATE TRIGGER `AggiornamentoCostoTotManutenzione`
AFTER INSERT ON `manutenzione` FOR EACH ROW
BEGIN
/* Se si tratta di una manutenzione svolta,
e non di una concimazione necessaria */
IF(NEW.`CodPianta` IS NOT NULL) THEN
UPDATE `specie`
SET `CostoTotManutenzione` = `CostoTotManutenzione` + NEW.Costo
WHERE `Nome` = (SELECT `NomeSpecie`
FROM `pianta`
WHERE `CodPianta` = NEW.`CodPianta`);
END IF;
END $$
DELIMITER ;

-- Controllo data scadenza
DROP TRIGGER IF EXISTS `ControllaDataScadenza`;
DELIMITER $$
CREATE TRIGGER `ControllaDataScadenza`
BEFORE INSERT ON `manutenzione` FOR EACH ROW
BEGIN
IF(NEW.`Scadenza` < NEW.`DataInizio`) THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'La scadenza non può avvenire
prima della data di inizio.';
END IF;
END $$
DELIMITER ;

--
-- Creazione tabella `media`
--
DROP TABLE IF EXISTS `media`;
CREATE TABLE `media` (
`URL` char(50) NOT NULL,
PRIMARY KEY (`URL`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `mediapost`
--
DROP TABLE IF EXISTS `mediapost`;
CREATE TABLE `mediapost` (
`Timestamp` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
ON UPDATE CURRENT_TIMESTAMP,
`Utente` char(50) NOT NULL,
`URL` char(50) NOT NULL,
PRIMARY KEY (`Timestamp`,`Utente`,`URL`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- Aggiornamento della ridondanza `NumMedia` dell'entità `thread`
DROP TRIGGER IF EXISTS `AggiornamentoNumMedia`;
CREATE TRIGGER `AggiornamentoNumMedia`
  AFTER INSERT ON `mediapost` FOR EACH ROW
  UPDATE `thread`
  SET `NumMedia` = `NumMedia` + 1
  WHERE `CodThread` = (SELECT `CodThread`
                      FROM `post`
                      WHERE `Timestamp` = NEW.`Timestamp`
                        AND `Utente` = NEW.`Utente`);

--
-- Creazione tabella `ordine`
--
DROP TABLE IF EXISTS `ordine`;
CREATE TABLE `ordine` (
  `CodOrdine` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `Timestamp` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
    ON UPDATE CURRENT_TIMESTAMP,
  `Stato` char(50) NOT NULL, /* pendente, in processazione,
                             in preparazione, spedito, evaso */
  `Utente` char(50) DEFAULT NULL,
  PRIMARY KEY (`CodOrdine`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `patologia`
--
DROP TABLE IF EXISTS `patologia`;
CREATE TABLE `patologia` (
  `CodPatologia` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `DataInizio` date NOT NULL,
  `DataFine` date NOT NULL,
  `Probabilita` float(13, 2) unsigned NOT NULL, -- percentuale
  `Entita` float(13, 2) NOT NULL, -- da 0 a 10
  PRIMARY KEY (`CodPatologia`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `patologiereport`
--
DROP TABLE IF EXISTS `patologiereport`;
CREATE TABLE `patologiereport` (
  `CodPianta` int(11) unsigned NOT NULL,
  `Timestamp` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
    ON UPDATE CURRENT_TIMESTAMP,
  `CodPatologia` int(11) unsigned NOT NULL,
  PRIMARY KEY (`CodPianta`, `Timestamp`, `CodPatologia`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `periodimanutenzione`
--
DROP TABLE IF EXISTS `periodimanutenzione`;
CREATE TABLE `periodimanutenzione` (
  `CodManutenzione` int(11) unsigned NOT NULL,
  `DataInizio` date NOT NULL,
  `DataFine` date NOT NULL,
  PRIMARY KEY (`CodManutenzione`, `DataInizio`, `DataFine`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

--
-- Creazione tabella `periodiprodotti`
--
DROP TABLE IF EXISTS `periodiprodotti`;
CREATE TABLE `periodiprodotti` (
  `NomeProdotto` char(50) NOT NULL,
  `DataInizio` date NOT NULL,
  `DataFine` date NOT NULL,
  PRIMARY KEY (`NomeProdotto`,`DataInizio`,`DataFine`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `periodispecie`
--
DROP TABLE IF EXISTS `periodispecie`;
CREATE TABLE `periodispecie` (
  `NomeSpecie` char(50) NOT NULL,
  `DataInizio` date NOT NULL,
  `DataFine` date NOT NULL,
  `Tipo` char(50) NOT NULL, -- fioritura, fruttificazione, riposo
  PRIMARY KEY (`NomeSpecie`,`DataInizio`,`DataFine`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `pianta`
--
DROP TABLE IF EXISTS `pianta`;
CREATE TABLE `pianta` (
  `CodPianta` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `DimAttuale` int(11) unsigned NOT NULL, -- cm
  `Prezzo` int(11) unsigned DEFAULT NULL, -- euro
  `IndiceManutenzione` float(13, 2) unsigned DEFAULT NULL,
  `NomeSpecie` char(50) DEFAULT NULL,
  `CodTerreno` int(11) unsigned NOT NULL,
  `CodContenitore` int(11) unsigned DEFAULT NULL,
  `CodOrdine` int(11) unsigned DEFAULT NULL,
  PRIMARY KEY (`CodPianta`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/* Stored procedure per le operazioni di aggiornamento delle ridondanze
e di controllo della capienza delle entità `Sezione`, `Serra` e `Sede` */
DROP PROCEDURE IF EXISTS AggiornaPiantePresentiSezioneSerraSede;
DELIMITER $$
CREATE PROCEDURE AggiornaPiantePresentiSezioneSerraSede(
  IN contenitore int(11),
  IN quantita int(1))
BEGIN
  -- Entità coinvolte
  SET @Sezione = (SELECT `CodSezione`
                  FROM `ripiano`
                  WHERE `CodRipiano` = (
                    SELECT `CodRipiano`
                    FROM `Contenitore`
                    WHERE `CodContenitore` = `contenitore`));
  SET @Serra = ( SELECT `CodSerra`
                 FROM `sezione`
                 WHERE `CodSezione` = @Sezione);

  SET @Sede = ( SELECT `CodSede`
                FROM `serra`
                WHERE `CodSerra` = @Serra);

```



```

-- Controllo capienze
IF(`quantita` > 0) THEN
    SET @PiantePresentiSezione = ( SELECT `PiantePresenti`
                                    FROM `sezione`
                                    WHERE `CodSezione` = @Sezione);
    SET @CapienzaSezione = (SELECT `Capienza`
                             FROM `sezione`
                             WHERE `CodSezione` = @Sezione);
    IF(@PiantePresentiSezione + `quantita` > @CapienzaSezione) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Superata capienza della sezione.';
    END IF;

    SET @PiantePresentiSerra = (SELECT `PiantePresenti`
                                  FROM `serra`
                                  WHERE `CodSerra` = @Serra);
    SET @CapienzaSerra = ( SELECT `Capienza`
                             FROM `serra`
                             WHERE `CodSerra` = @Serra);
    IF(@PiantePresentiSerra + `quantita` > @CapienzaSerra) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Superata capienza della serra.';
    END IF;

    SET @PiantePresentiSede = ( SELECT `PiantePresenti`
                                  FROM `sede`
                                  WHERE `CodSede` = @Sede);
    SET @CapienzaSede = ( SELECT `Capienza`
                             FROM `sede`
                             WHERE `CodSede` = @Sede);
    IF(@PiantePresentiSede + `quantita` > @CapienzaSede) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Superata capienza della sede.';
    END IF;
END IF;

-- Aggiornamento ridondanze
UPDATE `sezione`
SET `PiantePresenti` = `PiantePresenti` + `quantita`
WHERE `CodSezione` = @Sezione;

UPDATE `serra`
SET `PiantePresenti` = `PiantePresenti` + `quantita`
WHERE `CodSerra` = @Serra;

UPDATE `sede`
SET `PiantePresenti` = `PiantePresenti` + `quantita`
WHERE `CodSede` = @Sede;
END $$
DELIMITER ;

/* Stored procedure per l'aggiornamento della
ridondanza `NumeroPiante` dell'entità `specie` */
DROP PROCEDURE IF EXISTS AggiornaNumeroPianteSpecie;
CREATE PROCEDURE AggiornaNumeroPianteSpecie(
    IN pianta int(11),
    IN quantita int(1))

UPDATE `specie`
SET `NumeroPiante` = `NumeroPiante` + `quantita`
WHERE `Nome` = (SELECT `NomeSpecie`
                 FROM `pianta`
                 WHERE `CodPianta` = `pianta`);

```

```

-- Stored procedure per l'aggiornamento dello stato di un contenitore
DROP PROCEDURE IF EXISTS AggiornaStatoContenitore;
DELIMITER $$
CREATE PROCEDURE AggiornaStatoContenitore( IN contenitore int(11),
                                           IN nuovoStato tinyint(1))
BEGIN
-- Controllo dello stato del nuovo contenitore
IF((SELECT `Pieno`
     FROM `contenitore`
     WHERE `CodContenitore` = `contenitore`) = `nuovoStato`) THEN
IF(`nuovoStato` = '0') THEN
SET @messageText = 'Codice del vecchio contenitore errato.';
ELSEIF(`nuovoStato` = '1') THEN
SET @messageText = 'Impossibile inserire una nuova pianta
in un contenitore già pieno.';
END IF;

SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = @messageText;
END IF;

-- Aggiornamento stato
UPDATE `contenitore`
SET `Pieno` = `nuovoStato`
WHERE `CodContenitore` = `contenitore`;
END $$
DELIMITER ;

-- Stored procedure per il controllo della dimensione di un contenitore
DROP PROCEDURE IF EXISTS ControlloDimensioneContenitore;
DELIMITER $$
CREATE PROCEDURE ControlloDimensioneContenitore(IN dimAttuale int(11),
                                                IN contenitore int(11))
BEGIN
SET @DimensioneContenitore = ( SELECT `Dimensione`
                               FROM `contenitore`
                               WHERE `CodContenitore` = contenitore);

IF(`dimAttuale` > @DimensioneContenitore) THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'La pianta è attualmente troppo grande per
essere inserita nel contenitore';
END IF;
END $$
DELIMITER ;

-- Stored procedure per il controllo della temperatura di una sezione
DROP PROCEDURE IF EXISTS ControlloTemperaturaSezione;
DELIMITER $$
CREATE PROCEDURE ControlloTemperaturaSezione( IN pianta int(11),
                                              IN contenitore int(11))
BEGIN
SET @tempSezione = (
SELECT `Temperatura`
FROM `sezione`
WHERE `CodSezione` = (
SELECT `CodSezione`
FROM `ripiano`
WHERE `CodRipiano` = (
SELECT `CodRipiano`
FROM `contenitore`
WHERE `CodContenitore` = contenitore

```

```

    )
  )
);

SET @tempMin = (SELECT `TempMin`
                FROM `specie`
                WHERE `Nome` = (SELECT `NomeSpecie`
                                FROM `pianta`
                                WHERE `CodPianta` = pianta));

SET @tempMax = (SELECT `TempMax`
                FROM `specie`
                WHERE `Nome` = (SELECT `NomeSpecie`
                                FROM `pianta`
                                WHERE `CodPianta` = pianta));

IF(@tempSezione > @tempMax OR @tempSezione < @tempMin) THEN
  SIGNAL SQLSTATE '45000'
  SET MESSAGE_TEXT = 'La temperatura della sezione non e\'
  compatibile con questa pianta.';
END IF;
END $$
DELIMITER ;

-- Calcolo dell'indice di manutenzione
DROP TRIGGER IF EXISTS `ImpostaIndiceManutenzione`;
CREATE TRIGGER `ImpostaIndiceManutenzione`
  BEFORE INSERT ON `pianta` FOR EACH ROW
  SET NEW.`IndiceManutenzione` = (
    SELECT `IndiceAccrescimento` * `DimMax`
    FROM `specie`
    WHERE `Nome` = NEW.`NomeSpecie`) / NEW.`DimAttuale`;

-- Trigger per l'inserimento di una nuova pianta
DROP TRIGGER IF EXISTS `InserimentoPianta`;
DELIMITER $$
CREATE TRIGGER `InserimentoPianta` AFTER INSERT ON `pianta` FOR EACH ROW
BEGIN
  -- Se la pianta e' contenuta in un contenitore
  IF(NEW.`CodContenitore` IS NOT NULL) THEN
    CALL ControlloDimensioneContenitore(NEW.`DimAttuale`,
                                        NEW.`CodContenitore`);
    CALL ControlloTemperaturaSezione(NEW.`CodPianta`,
                                      NEW.`CodContenitore`);
    CALL AggiornaStatoContenitore(NEW.`CodContenitore`,
                                  '1');
    CALL AggiornaPiantePresentiSezioneSerraSede(NEW.`CodContenitore`,
                                                  '1');
  END IF;
  CALL AggiornaNumeroPianteSpecie(NEW.`CodPianta`, '1');
END $$
DELIMITER ;

-- Trigger per la modifica di una pianta
DROP TRIGGER IF EXISTS `ModificaPianta`;
DELIMITER $$
CREATE TRIGGER `ModificaPianta` BEFORE UPDATE ON `pianta` FOR EACH ROW
BEGIN
  IF(OLD.`CodContenitore` <> NEW.`CodContenitore`) THEN
    IF(OLD.`CodContenitore` IS NOT NULL) THEN
      CALL AggiornaStatoContenitore(OLD.`CodContenitore`,
                                    '0');
    END IF;
  END IF;
END $$
DELIMITER ;

```

```

        CALL AggiornaPiantePresentiSezioneSerraSede(
            OLD.`CodContenitore`,
            '-1'
        );
    END IF;
    IF(NEW.`CodContenitore` IS NOT NULL) THEN
        CALL ControlloDimensioneContenitore(NEW.`DimAttuale`,
            NEW.`CodContenitore`);
        CALL AggiornaStatoContenitore(NEW.`CodContenitore`, '1');
        CALL AggiornaPiantePresentiSezioneSerraSede(
            NEW.`CodContenitore`,
            '1'
        );
    END IF;
END IF;
END $$
DELIMITER ;

-- Trigger per la cancellazione di una pianta
DROP TRIGGER IF EXISTS `CancellazionePianta`;
DELIMITER $$
CREATE TRIGGER `CancellazionePianta` AFTER DELETE ON `pianta` FOR EACH ROW
BEGIN
    CALL AggiornaStatoContenitore(OLD.`CodContenitore`, '0');
    CALL AggiornaPiantePresentiSezioneSerraSede(OLD.`CodContenitore`, '-1');
    CALL AggiornaNumeroPianteSpecie(OLD.`CodPianta`, '-1');
END $$
DELIMITER ;

--
-- Creazione tabella `piantearredamentoinpienaterra`
--
DROP TABLE IF EXISTS `piantearredamentoinpienaterra`;
CREATE TABLE `piantearredamentoinpienaterra` (
    `CodPianta` int(11) unsigned NOT NULL,
    `CodSpazio` int(11) unsigned NOT NULL,
    `Versione` int(11) unsigned NOT NULL,
    `PosX` int(11) NOT NULL,
    `PosY` int(11) NOT NULL,
    PRIMARY KEY (`CodPianta`,`CodSpazio`,`Versione`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `post`
--
DROP TABLE IF EXISTS `post`;
CREATE TABLE `post` (
    `Timestamp` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
    ON UPDATE CURRENT_TIMESTAMP,
    `Utente` char(50) DEFAULT 'Utente eliminato',
    `Giudizio` float(2, 1) DEFAULT NULL, -- da 0 a 5
    `Testo` char(50) NOT NULL,
    `CodThread` int(11) unsigned NOT NULL,
    PRIMARY KEY (`Timestamp`,`Utente`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Aggiornamento della ridondanza `NumPostPubblicati` dell'entità `account`
DROP TRIGGER IF EXISTS `AggiornamentoNumPostPubblicati`;
CREATE TRIGGER `AggiornamentoNumPostPubblicati`
AFTER INSERT ON `post` FOR EACH ROW
UPDATE `account`
SET `NumPostPubblicati` = `NumPostPubblicati` + 1

```

```

WHERE `Nickname` = NEW.`Utente`;

--
-- Creazione tabella `principiattiviprodotto`
--
DROP TABLE IF EXISTS `principiattiviprodotto`;
CREATE TABLE `principiattiviprodotto` (
  `NomeProdotto` char(50) NOT NULL,
  `NomePrincipioAttivo` char(50) NOT NULL,
  `Concentrazione` float(13, 2) unsigned NOT NULL, -- mg/m^3
  PRIMARY KEY (`NomeProdotto`,`NomePrincipioAttivo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `principioattivo`
--
DROP TABLE IF EXISTS `principioattivo`;
CREATE TABLE `principioattivo` (
  `Nome` char(50) NOT NULL,
  PRIMARY KEY (`Nome`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `prodottipatologia`
--
DROP TABLE IF EXISTS `prodottipatologia`;
CREATE TABLE `prodottipatologia` (
  `CodPatologia` int(11) unsigned NOT NULL,
  `NomeProdotto` char(50) NOT NULL,
  PRIMARY KEY (`CodPatologia`,`NomeProdotto`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `prodottitrattamento`
--
DROP TABLE IF EXISTS `prodottitrattamento`;
CREATE TABLE `prodottitrattamento` (
  `NomeProdotto` char(50) NOT NULL,
  `CodManutenzione` int(11) unsigned NOT NULL,
  `Dose` float(13, 2) NOT NULL, -- mg/m^3
  PRIMARY KEY (`NomeProdotto`,`CodManutenzione`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `prodotto`
--
DROP TABLE IF EXISTS `prodotto`;
CREATE TABLE `prodotto` (
  `Nome` char(50) NOT NULL,
  `Marca` char(50) NOT NULL,
  `TempoMinConsumazFrutti` int(11) DEFAULT '0', -- giorni
  `ModalitaSomm` char(50) DEFAULT NULL, -- disciolto, nebulizzato
  PRIMARY KEY (`Nome`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `prodottocombatte`
--
DROP TABLE IF EXISTS `prodottocombatte`;
CREATE TABLE `prodottocombatte` (
  `NomeProdotto` char(50) NOT NULL,
  `NomeAgentePatogeno` char(50) NOT NULL,

```

```

`Dosaggio` float(13, 2) unsigned NOT NULL, -- mL
PRIMARY KEY (`NomeProdotto`,`NomeAgentePatogeno`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `reportassunzioni`
--
DROP TABLE IF EXISTS `reportassunzioni`;
CREATE TABLE `reportassunzioni` (
  `CodReportAss` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `CodSede` int(11) unsigned NOT NULL,
  `NumDipendenti` int(11) unsigned NOT NULL,
  `Indeterminato` tinyint(1) DEFAULT '0',
  `DataInizio` date DEFAULT NULL,
  `DataFine` date DEFAULT NULL,
  PRIMARY KEY (`CodReportAss`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Creazione dei reports delle assunzioni
DROP EVENT IF EXISTS `CreaReportAssunzioni`;
DELIMITER $$
CREATE EVENT `CreaReportAssunzioni` ON SCHEDULE EVERY 1 MONTH
STARTS '2014-01-01 23:55:00' DO
BEGIN
  SET @massimoNumeroOrdiniGestibiliDaUnDipendente = '20';

  /* Determina per quante volte una sede puo' assumere dipendenti
  part-time prima di iniziare ad assumere a tempo indeterminato */
  SET @numeroDiAssunzioniPrimaDiIndeterminato = '5';

  CREATE OR REPLACE VIEW `NuoviOrdiniPerSede` AS
  SELECT `CodSede`, COUNT(*) AS `OrdiniUltimoMeseDaGestire`
  FROM `serra`
  WHERE `CodSerra` IN (
    SELECT DISTINCT(`CodSerra`)
    FROM `sezione`
    WHERE `CodSezione` IN (
      SELECT DISTINCT(`CodSezione`)
      FROM `ripiano`
      WHERE `CodRipiano` IN (
        SELECT DISTINCT(`CodRipiano`)
        FROM `contenitore`
        WHERE `CodContenitore` IN (
          SELECT DISTINCT(`CodContenitore`)
          FROM `pianta`
          WHERE `CodOrdine` IS NOT NULL
          AND `CodContenitore` IS NOT NULL
          AND `CodOrdine` IN (
            SELECT `CodOrdine`
            FROM `ordine`
            WHERE `Timestamp` BETWEEN
              CURRENT_TIMESTAMP - INTERVAL 1 MONTH
              AND
              CURRENT_TIMESTAMP
            AND (`Stato` = 'pendente' OR
              `Stato` = 'in processazione')
          )
        )
      )
    )
  )
  GROUP BY `CodSede`;

```

```

CREATE OR REPLACE VIEW `RapportoNuoviOrdiniSuDipendentiPerSede` AS
  SELECT  NOPS.`CodSede`,
          `NumDipendenti`/`OrdiniUltimoMeseDaGestire`
          AS `RapportoOrdiniDipendenti`,
        (
          SELECT COUNT(*)
          FROM `reportassunzioni` RA
          WHERE RA.`CodSede` = NOPS.`CodSede`)
          AS `NumeroReportPrecedenti`
  FROM `NuoviOrdiniPerSede` NOPS NATURAL JOIN `Sede` S;

INSERT INTO `reportassunzioni` (`NumDipendenti`,
                                `CodSede`,
                                `Indeterminato`,
                                `DataInizio`,
                                `DataFine`)

  SELECT  RNO.`CodSede`,
          FLOOR(RNO.`RapportoOrdiniDipendenti`/20)
          AS `DipendentiDaAssumere`,
          IF((SELECT RA1.`Indeterminato` AS `IndeterminatoUltimoReport`
              FROM `reportassunzioni` RA1
              WHERE RA1.`CodSede` = RNO.`CodSede`
              AND RA1.`CodReportAss` = (
                SELECT MAX(`CodReportAss`)
                FROM `reportassunzioni` RA2
                WHERE RA2.`CodSede` = RA1.`CodSede`)) = '1',
            '0',
            IF(RNO.`NumeroReportPrecedenti` <
              @numeroDiAssunzioniPrimaDiIndeterminato, '0', '1')
            ) AS `Indeterminato`,
          CURRENT_DATE AS `DataInizio`,
          CURRENT_DATE + INTERVAL 1 MONTH AS `DataFine`
  FROM `RapportoNuoviOrdiniSuDipendentiPerSede` RNO
  WHERE RNO.`RapportoOrdiniDipendenti` >
        @massimoNumeroOrdiniGestibiliDaUnDipendente;

END $$
DELIMITER ;

--
-- Creazione tabella `reportdiagnostica`
--
DROP TABLE IF EXISTS `reportdiagnostica`;
CREATE TABLE `reportdiagnostica` (
  `CodPianta` int(11) unsigned NOT NULL,
  `Timestamp` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
  ON UPDATE CURRENT_TIMESTAMP,
  `CodTerreno` int(11) unsigned DEFAULT NULL,
  PRIMARY KEY (`Timestamp`,`CodPianta`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Impostazione dell'attributo `CodTerreno`
DROP TRIGGER IF EXISTS `ImpostaCodTerreno`;
CREATE TRIGGER `ImpostaCodTerreno`
  BEFORE INSERT ON `reportdiagnostica` FOR EACH ROW
  SET NEW.`CodTerreno` = (SELECT `CodTerreno`
                        FROM `pianta`
                        WHERE `CodPianta` = NEW.`CodPianta`);

-- Aggiornamento della ridondanza `NumeroEsordi` dell'entità `specie`
DROP TRIGGER IF EXISTS `AggiornamentoNumeroEsordi`;
CREATE TRIGGER `AggiornamentoNumeroEsordi`
  AFTER INSERT ON `reportdiagnostica` FOR EACH ROW

```

```

UPDATE `specie`
SET `NumeroEsordi` = `NumeroEsordi` + 1
WHERE `Nome` = (SELECT `NomeSpecie`
                FROM `pianta`
                WHERE `CodPianta` = NEW.`CodPianta`);

--
-- Creazione tabella `reportmanutenzione`
--
DROP TABLE IF EXISTS `reportmanutenzione`;
CREATE TABLE `reportmanutenzione` (
  `Tipo` char(50) NOT NULL,
  `NumSnapshot` int(11) unsigned NOT NULL DEFAULT '1',
  `NomeSpecie` char(50) NOT NULL,
  `CostoTot` int(11) unsigned NOT NULL DEFAULT '0',
  PRIMARY KEY (`Tipo`,`NumSnapshot`,`NomeSpecie`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Creazione dei reports di manutenzione
DROP EVENT IF EXISTS `CreaReportManutenzione`;
DELIMITER $$
CREATE EVENT `CreaReportManutenzione` ON SCHEDULE EVERY 3 MONTH
STARTS '2014-01-01 23:55:00' DO
BEGIN
  INSERT INTO `reportmanutenzione`
    SELECT M.`TipoManutenzione` AS `Tipo`,
           (SELECT COUNT(*)
            FROM `reportmanutenzione` RM
            WHERE RM.`NomeSpecie` = P.`NomeSpecie`
                  AND RM.`Tipo` = M.`TipoManutenzione`) + 1
           AS `NumSnapshot`,
           P.`NomeSpecie`,
           SUM(M.`Costo`) AS `CostoTot`
    FROM `manutenzione` M INNER JOIN
         (SELECT `CodPianta`,`NomeSpecie` FROM `pianta`) P
         ON M.`CodPianta` = P.`CodPianta`
    -- manutenzioni svolte, non solo esatte
    WHERE M.`CodPianta` IS NOT NULL
          AND M.`DataInizio` BETWEEN -- degli ultimi 3 mesi
          CURRENT_DATE - INTERVAL 3 MONTH
          AND
          CURRENT_DATE
    GROUP BY P.`NomeSpecie`, M.`TipoManutenzione`;
END $$
DELIMITER ;

--
-- Creazione tabella `reportordini`
--
DROP TABLE IF EXISTS `reportordini`;
CREATE TABLE `reportordini` (
  `CodRepOrdini` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `DaOrdinare` tinyint(1) NOT NULL,
  `Clima` char(50) DEFAULT NULL,
  PRIMARY KEY (`CodRepOrdini`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Creazione dei reports degli ordini da effettuare
DROP EVENT IF EXISTS `CreaReportOrdiniDaEffettuare`;
DELIMITER $$
CREATE EVENT `CreaReportOrdiniDaEffettuare` ON SCHEDULE EVERY 1 WEEK
STARTS '2014-01-01 23:55:00' DO

```



```

BEGIN
CREATE OR REPLACE VIEW `LucePerSpecie` AS
SELECT S.`Nome` AS `NomeSpecie`,
      IF( PS.`Tipo` = 'riposo',
         S.`OreLuceRiposo`,
         S.`OreLuceVegetativo`) AS `OreLuce`,
      IF( PS.`Tipo` = 'riposo',
         'riposo',
         'vegetativo') AS `TipoPeriodo`,
      DATEDIFF(PS.`DataFine`, PS.`DataInizio`) AS `NumeroGiorni`
FROM `specie` S INNER JOIN `periodispecie` PS
ON S.`Nome` = PS.`NomeSpecie`;

CREATE OR REPLACE VIEW `OreLuceMediePerSpecie` AS
SELECT `NomeSpecie`,
      (SUM(`OreLuce` * `NumeroGiorni`))/365 AS `OreLuceMedie`
FROM ( SELECT `NomeSpecie`,
              `TipoPeriodo`,
              `OreLuce`,
              SUM(`NumeroGiorni`) AS `NumeroGiorni`
        FROM `LucePerSpecie`
        GROUP BY `NomeSpecie`, `TipoPeriodo`) AS D
GROUP BY `NomeSpecie`;

CREATE OR REPLACE VIEW `PianteDaOrdinare` AS
SELECT P.`NomeSpecie`,
      COUNT(*) AS `PianteDaOrdinare`
FROM `ordine` O
INNER JOIN
`pianta` P ON O.`CodOrdine` = P.`CodOrdine`
INNER JOIN
`specie` S ON P.`NomeSpecie` = S.`Nome`
WHERE P.`CodOrdine` IS NOT NULL
      AND O.`Stato` = 'pendente'
GROUP BY P.`NomeSpecie`;

CREATE OR REPLACE VIEW `PianteEstiveDaOrdinare` AS
SELECT `NomeSpecie`, `PianteDaOrdinare`
FROM `PianteDaOrdinare` NATURAL JOIN `OreLuceMediePerSpecie`
WHERE `OreLuceMedie` > '7.5';

CREATE OR REPLACE VIEW `PianteInvernaliDaOrdinare` AS
SELECT `NomeSpecie`, `PianteDaOrdinare`
FROM `PianteDaOrdinare` NATURAL JOIN `OreLuceMediePerSpecie`
WHERE `OreLuceMedie` <= '7.5';

INSERT INTO `reportordini` (`DaOrdinare`, `Clima`)
SELECT '1', 'estivo'
FROM `PianteEstiveDaOrdinare`;

INSERT INTO `speciereportordini` ( `CodRepOrdini`,
                                  `NomeSpecie`,
                                  `Quantita`)
SELECT ( SELECT MAX(`CodRepOrdini`)
          FROM `reportordini`) AS `CodRepOrdini`,
       `NomeSpecie`,
       `PianteDaOrdinare`
FROM `PianteEstiveDaOrdinare`;

INSERT INTO `reportordini` (`DaOrdinare`, `Clima`)
SELECT '1', 'invernale'
FROM `PianteInvernaliDaOrdinare`;

```

```

INSERT INTO `speciereportordini` ( `CodRepOrdini`,
                                   `NomeSpecie`,
                                   `Quantita`)
SELECT ( SELECT MAX(`CodRepOrdini`)
         FROM `reportordini`) AS `CodRepOrdini`,
       `NomeSpecie`,
       `PianteDaOrdinare`
FROM `PianteInvernalidiDaOrdinare`;
END $$
DELIMITER ;

-- Creazione dei reports degli ordini da non effettuare
DROP EVENT IF EXISTS `CreaReportOrdiniDaNonEffettuare`;
DELIMITER $$
CREATE EVENT `CreaReportOrdiniDaNonEffettuare` ON SCHEDULE EVERY 1 YEAR
STARTS '2014-01-01 23:55:00' DO
BEGIN

END $$
DELIMITER ;

--
-- Creazione tabella `ripiano`
--
DROP TABLE IF EXISTS `ripiano`;
CREATE TABLE `ripiano` (
  `CodRipiano` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `Irrigazione` float (13, 2) NOT NULL, -- percentuale
  `CodSezione` int(11) unsigned DEFAULT NULL,
  `Capienza` int(11) unsigned NOT NULL,
  PRIMARY KEY (`CodRipiano`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Controllo sulla capienza di un ripiano
DROP TRIGGER IF EXISTS `ControllaCapienzaRipiano`;
DELIMITER $$
CREATE TRIGGER `ControllaCapienzaRipiano`
BEFORE INSERT ON `ripiano` FOR EACH ROW
BEGIN
  SET @CapienzaSezione = (SELECT `Capienza`
                        FROM `sezione`
                        WHERE `CodSezione` = NEW.`CodSezione`);

  SET @CapienzaRipiani = (SELECT SUM(`Capienza`)
                        FROM `ripiano`
                        WHERE `CodSezione` = NEW.`CodSezione`);

  IF(@CapienzaRipiani + NEW.`Capienza` > @CapienzaSezione) THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'La capienza di tutti i ripiani di una
    sezione non può superare quella di quest\'ultima.';
  END IF;
END $$
DELIMITER ;

--
-- Creazione tabella `salute`
--
DROP TABLE IF EXISTS `salute`;
CREATE TABLE `salute` (
  `CodContentitore` int(11) unsigned NOT NULL,

```

```

`Timestamp` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
ON UPDATE CURRENT_TIMESTAMP,
`Umidita` float(13, 2) NOT NULL, -- percentuale
`TassoAmmoniaca` float(13, 2) NOT NULL, -- percentuale
`LivelloGas` float(13, 2) NOT NULL, -- percentuale
PRIMARY KEY (`Timestamp`,`CodContentitore`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `scheda`
--
DROP TABLE IF EXISTS `scheda`;
CREATE TABLE `scheda` (
  `CodScheda` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `DataAcquisto` date DEFAULT NULL,
  `Settore` int(11) unsigned DEFAULT NULL,
  `Collocazione` char(50) NOT NULL DEFAULT 'piena terra', /* piena terra,
                                                             vaso */
  `DimensioneAllAcquisto` int(11) unsigned DEFAULT NULL,
  `PosX` int(11) NOT NULL,
  `PosY` int(11) NOT NULL,
  `CodPianta` int(11) unsigned NOT NULL,
  `CodVaso` int(11) unsigned DEFAULT NULL,
  `Utente` char(50) NOT NULL,
  PRIMARY KEY (`CodScheda`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Impostazione dell'attributo `DimensioneAllAcquisto`
DROP TRIGGER IF EXISTS `ImpostazioneDimensioneAllAcquisto`;
CREATE TRIGGER `ImpostazioneDimensioneAllAcquisto`
  BEFORE INSERT ON `scheda` FOR EACH ROW
  SET NEW.`DimensioneAllAcquisto` = (
    SELECT `DimAttuale`
    FROM `pianta`
    WHERE `CodPianta` = NEW.`CodPianta`);

/* Trigger per l'aggiornamento della ridondanza
`NumPianteVendute` di `specie` dopo la vendita di una pianta */
DROP TRIGGER IF EXISTS `AggiornaNumPianteVendute`;
DELIMITER $$
CREATE TRIGGER `AggiornaNumPianteVendute`
  AFTER INSERT ON `scheda` FOR EACH ROW
BEGIN
  UPDATE `specie`
  SET `NumPianteVendute` = `NumPianteVendute` + 1
  WHERE `Nome` = (SELECT `NomeSpecie`
                  FROM `pianta`
                  WHERE `CodPianta` = NEW.`CodPianta`);
END $$
DELIMITER ;

--
-- Creazione tabella `sede`
--
DROP TABLE IF EXISTS `sede`;
CREATE TABLE `sede` (
  `CodSede` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `NumDipendenti` int(11) unsigned NOT NULL,
  `Indirizzo` char(50) NOT NULL,
  `PiantePresenti` int(11) NOT NULL DEFAULT '0',
  `Nome` char(50) NOT NULL,
  `Capienza` int(11) unsigned NOT NULL,

```

```

    PRIMARY KEY (`CodSede`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `serra`
--
DROP TABLE IF EXISTS `serra`;
CREATE TABLE `serra` (
  `CodSerra` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `Nome` char(50) NOT NULL,
  `Indirizzo` char(50) NOT NULL,
  `Larghezza` int(11) unsigned NOT NULL, -- metri
  `Altezza` int(11) unsigned NOT NULL, -- metri
  `Capienza` int(11) unsigned NOT NULL,
  `PiantePresenti` int(11) unsigned NOT NULL DEFAULT '0',
  `CodSede` int(11) unsigned NOT NULL,
  PRIMARY KEY (`CodSerra`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Controllo sulla capienza di una serra
DROP TRIGGER IF EXISTS `ControllaCapienzaSerra`;
DELIMITER $$
CREATE TRIGGER `ControllaCapienzaSerra`
  BEFORE INSERT ON `serra` FOR EACH ROW
BEGIN
  SET @CapienzaSede = ( SELECT `Capienza`
                        FROM `sede`
                        WHERE `CodSede` = NEW.`CodSede`);

  SET @CapienzaSerre = ( SELECT SUM(`Capienza`)
                        FROM `serra`
                        WHERE `CodSede` = NEW.`CodSede`);

  IF(@CapienzaSerre + NEW.`Capienza` > @CapienzaSede) THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'La capienza di tutte le serre di una
sede non può superare quella di quest\'ultima.';
  END IF;
END $$
DELIMITER ;

--
-- Creazione tabella `settore`
--
DROP TABLE IF EXISTS `settore`;
CREATE TABLE `settore` (
  `CodSettore` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `Tipo` char(50) NOT NULL, -- piena terra o pavimentato
  `Esposizione` char(50) NOT NULL, -- punti cardinali
  `NumOreLuce` int(11) unsigned NOT NULL,
  `CodSpazio` int(11) unsigned NOT NULL,
  PRIMARY KEY (`CodSettore`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `sezione`
--
DROP TABLE IF EXISTS `sezione`;
CREATE TABLE `sezione` (
  `CodSezione` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `Capienza` int(11) unsigned NOT NULL,
  `Irrigazione` float (13, 2) NOT NULL, -- percentuale

```

```

`Illuminazione` float (13, 2) NOT NULL, -- percentuale
`Umidita` float (13, 2) NOT NULL, -- percentuale
`Temperatura` float (13, 2) NOT NULL, -- gradi centigradi
`Quarantena` tinyint(1) NOT NULL,
`PiantePresenti` int(11) unsigned NOT NULL DEFAULT '0',
`Nome` char(50) NOT NULL,
`CodSerra` int(11) unsigned NOT NULL,
PRIMARY KEY (`CodSezione`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Controllo sulla capienza di una sezione
DROP TRIGGER IF EXISTS `ControllaCapienzaSezione`;
DELIMITER $$
CREATE TRIGGER `ControllaCapienzaSezione`
  BEFORE INSERT ON `sezione` FOR EACH ROW
BEGIN
  SET @CapienzaSerra = ( SELECT `Capienza`
                        FROM `serra`
                        WHERE `CodSerra` = NEW.`CodSerra`);

  SET @CapienzaSezioni = (SELECT SUM(`Capienza`)
                        FROM `sezione`
                        WHERE `CodSerra` = NEW.`CodSerra`);

  IF(@CapienzaSezioni + NEW.`Capienza` > @CapienzaSerra) THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'La capienza di tutte le sezioni di
      una serra non può superare quella di quest\'ultima.';
  END IF;
END $$
DELIMITER ;

--
-- Creazione tabella `sintomipatologia`
--
DROP TABLE IF EXISTS `sintomipatologia`;
CREATE TABLE `sintomipatologia` (
  `CodPatologia` int(11) unsigned NOT NULL,
  `CodSintomo` int(11) unsigned NOT NULL,
  PRIMARY KEY (`CodPatologia`,`CodSintomo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `sintomireport`
--
DROP TABLE IF EXISTS `sintomireport`;
CREATE TABLE `sintomireport` (
  `CodPianta` int(11) unsigned NOT NULL,
  `Timestamp` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
    ON UPDATE CURRENT_TIMESTAMP,
  `CodSintomo` int(11) unsigned NOT NULL,
  PRIMARY KEY (`CodPianta`,`Timestamp`,`CodSintomo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `sintomo`
--
DROP TABLE IF EXISTS `sintomo`;
CREATE TABLE `sintomo` (
  `CodSintomo` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `Descrizione` char(200) DEFAULT NULL,
  PRIMARY KEY (`CodSintomo`)

```

```

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `somministrazioneconcimazione`
--
DROP TABLE IF EXISTS `somministrazioneconcimazione`;
CREATE TABLE `somministrazioneconcimazione` (
  `NomeElemento` char(50) NOT NULL,
  `CodManutenzione` int(11) unsigned NOT NULL,
  `Iterazione` int(11) unsigned NOT NULL,
  `Quantita` float(13, 2) unsigned NOT NULL, -- mg/m^3
  PRIMARY KEY (`NomeElemento`,`CodManutenzione`,`Iterazione`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `spazio`
--
DROP TABLE IF EXISTS `spazio`;
CREATE TABLE `spazio` (
  `CodSpazio` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `Tipo` char(50) NOT NULL,
  `Utente` char(50) DEFAULT NULL,
  PRIMARY KEY (`CodSpazio`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `specie`
--
DROP TABLE IF EXISTS `specie`;
CREATE TABLE `specie` (
  `Nome` char(50) NOT NULL,
  `Genere` char(50) NOT NULL,
  `Cultivar` char(50) NOT NULL,
  `Infestante` tinyint(1) NOT NULL DEFAULT '0',
  `DimMax` int(11) unsigned NOT NULL, -- centimetri (diametro)
  `IndiceAccrescimento` float(13, 2) NOT NULL, -- di solito da 0 a 20
  `PosLuce` char(50) NOT NULL, -- pieno sole, ombra o mezz'ombra
  `TipoLuce` char(50) NOT NULL, -- in/diretta
  `OreLuceVegetativo` int(11) NOT NULL, -- giornaliera
  `TempMax` int(11) NOT NULL, -- gradi centigradi
  `ConsistenzaTerreno` char(50) DEFAULT NULL,
  `DistanzaMinConflitto` int(11) unsigned DEFAULT NULL, -- centimetri
  `NumeroEsordi` int(11) unsigned DEFAULT '0',
  `NumeroPianta` int(11) unsigned DEFAULT '0',
  `CostoTotManutenzione` int(11) unsigned DEFAULT '0', -- euro
  `Dioica` tinyint(1) NOT NULL DEFAULT '0',
  `TempMin` int(11) NOT NULL, -- gradi centigradi
  `OreLuceRiposo` int(11) NOT NULL, -- giornaliera
  `Fogliame` char(50) NOT NULL,
  `NumIrrigGiornalieraVegetativo` int(11) NOT NULL,
  `NumIrrigGiornalieraRiposo` int(11) NOT NULL,
  `QuantitaIrrigVegetativo` int(11) NOT NULL, -- mL
  `QuantitaIrrigRiposo` int(11) NOT NULL, -- mL
  `PeriodicitaIrrigVegetativo` int(11) NOT NULL, -- ogni quanti giorni
  `PeriodicitaIrrigRiposo` int(11) NOT NULL, -- ogni quanti giorni
  `NumPiantaVendute` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`Nome`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Calcolo dell'indice di accrescimento
DROP TRIGGER IF EXISTS `ImpostaIndiceAccrescimento`;
CREATE TRIGGER `ImpostaIndiceAccrescimento`

```

```

BEFORE INSERT ON `specie` FOR EACH ROW
SET NEW.`IndiceAccrescimento` = NEW.`DimMax` * 100 /
  (NEW.`NumIrrigGiornaliereVegetativo` *
    NEW.`QuantitaIrrigVegetativo` *
    365/NEW.`PeriodicitaIrrigVegetativo`
  + 0.5 * (NEW.`NumIrrigGiornaliereRiposo` *
    NEW.`QuantitaIrrigRiposo` *
    365/NEW.`PeriodicitaIrrigRiposo`));

-- Controllo sulla distanza minima di conflitto
DROP TRIGGER IF EXISTS `ControllaDistanzaMinConflitto`;
DELIMITER $$
CREATE TRIGGER `ControllaDistanzaMinConflitto`
  BEFORE INSERT ON `specie` FOR EACH ROW
BEGIN
  IF(NEW.`DistanzaMinConflitto` < NEW.`DimMax` / 2) THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'DistanzaMinConflitto troppo piccola.';
  END IF;
END $$
DELIMITER ;

-- Controllo del range delle temperature accettabili
DROP TRIGGER IF EXISTS `ControllaTemperature`;
DELIMITER $$
CREATE TRIGGER `ControllaTemperature`
  BEFORE INSERT ON `specie` FOR EACH ROW
BEGIN
  IF(NEW.`TempMin` > NEW.`TempMax`) THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Range di temperature non valido.';
  END IF;
END $$
DELIMITER ;

--
-- Creazione tabella `specieappassionatoaccount`
--
DROP TABLE IF EXISTS `specieappassionatoaccount`;
CREATE TABLE `specieappassionatoaccount` (
  `NomeSpecie` char(50) NOT NULL,
  `Utente` char(50) NOT NULL,
  PRIMARY KEY (`NomeSpecie`,`Utente`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `speciereportordini`
--
DROP TABLE IF EXISTS `speciereportordini`;
CREATE TABLE `speciereportordini` (
  `CodRepOrdini` int(11) unsigned NOT NULL,
  `NomeSpecie` char(50) NOT NULL,
  `Quantita` int(11) unsigned NOT NULL,
  PRIMARY KEY (`CodRepOrdini`,`NomeSpecie`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `terreno`
--
DROP TABLE IF EXISTS `terreno`;
CREATE TABLE `terreno` (
  `CodTerreno` int(11) unsigned NOT NULL AUTO_INCREMENT,

```

```

`PH` int(11) unsigned NOT NULL, -- da 3.5 a 9.0
`Consistenza` char(50) NOT NULL,
-- millidarcy (en.wikipedia.org/wiki/Permeability_(earth_sciences)#Units)
`Permeabilita` float(20, 10) NOT NULL,
PRIMARY KEY (`CodTerreno`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `thread`
--
DROP TABLE IF EXISTS `thread`;
CREATE TABLE `thread` (
  `CodThread` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `Nome` char(50) NOT NULL,
  `NumMedia` int(11) unsigned NOT NULL DEFAULT '0',
  PRIMARY KEY (`CodThread`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `vasiarredamento`
--
DROP TABLE IF EXISTS `vasiarredamento`;
CREATE TABLE `vasiarredamento` (
  `CodVaso` int(11) unsigned NOT NULL,
  `CodSpazio` int(11) unsigned NOT NULL,
  `Versione` int(11) unsigned NOT NULL,
  `CodPianta` int(11) unsigned NOT NULL,
  `PosX` int(11) NOT NULL,
  `PosY` int(11) NOT NULL,
  PRIMARY KEY (`CodVaso`,`CodSpazio`,`Versione`,`CodPianta`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `vaso`
--
DROP TABLE IF EXISTS `vaso`;
CREATE TABLE `vaso` (
  `CodVaso` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `Materiale` char(50) NOT NULL,
  `DimensioneX` int(11) unsigned NOT NULL,
  `DimensioneY` int(11) unsigned NOT NULL,
  PRIMARY KEY (`CodVaso`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `vertice`
--
DROP TABLE IF EXISTS `vertice`;
CREATE TABLE `vertice` (
  `PosX` int(11) NOT NULL,
  `PosY` int(11) NOT NULL,
  PRIMARY KEY (`PosX`,`PosY`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Creazione tabella `verticisetto`
--
DROP TABLE IF EXISTS `verticisetto`;
CREATE TABLE `verticisetto` (
  `CodSetto` int(11) unsigned NOT NULL,
  `PosX` int(11) NOT NULL,
  `PosY` int(11) NOT NULL,

```



```

    PRIMARY KEY (`CodSettore`,`PosX`,`PosY`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Foreign keys
--
ALTER TABLE `pianta`
  ADD FOREIGN KEY(`NomeSpecie`) REFERENCES `specie`(`Nome`)
    ON DELETE SET NULL ON UPDATE CASCADE,
  ADD FOREIGN KEY(`CodTerreno`) REFERENCES `terreno`(`CodTerreno`)
    ON DELETE CASCADE ON UPDATE CASCADE,
  ADD FOREIGN KEY(`CodOrdine`) REFERENCES `ordine`(`CodOrdine`)
    ON DELETE SET NULL ON UPDATE CASCADE,
  ADD FOREIGN KEY(`CodContenitore`)
    REFERENCES `contenitore`(`CodContenitore`)
    ON DELETE SET NULL ON UPDATE CASCADE;

ALTER TABLE `scheda`
  ADD FOREIGN KEY(`CodPianta`) REFERENCES `pianta`(`CodPianta`)
    ON DELETE CASCADE ON UPDATE CASCADE,
  ADD FOREIGN KEY(`CodVaso`) REFERENCES `vaso`(`CodVaso`)
    ON DELETE SET NULL ON UPDATE CASCADE,
  ADD FOREIGN KEY(`Utente`) REFERENCES `account`(`Nickname`)
    ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `ordine`
  ADD FOREIGN KEY(`Utente`) REFERENCES `account`(`Nickname`)
    ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `post`
  ADD FOREIGN KEY(`Utente`) REFERENCES `account`(`Nickname`)
    ON DELETE CASCADE ON UPDATE CASCADE,
  ADD FOREIGN KEY(`CodThread`) REFERENCES `thread`(`CodThread`)
    ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `mediapost`
  ADD FOREIGN KEY(`Timestamp`,`Utente`)
    REFERENCES `post`(`Timestamp`,`Utente`)
    ON DELETE CASCADE ON UPDATE CASCADE,
  ADD FOREIGN KEY(`URL`) REFERENCES `media`(`URL`)
    ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `specieappassionatoaccount`
  ADD FOREIGN KEY(`NomeSpecie`) REFERENCES `specie`(`Nome`)
    ON DELETE CASCADE ON UPDATE CASCADE,
  ADD FOREIGN KEY(`Utente`) REFERENCES `account`(`Nickname`)
    ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `reportmanutenzione`
  ADD FOREIGN KEY(`NomeSpecie`) REFERENCES `specie`(`Nome`)
    ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `speciereportordini`
  ADD FOREIGN KEY(`CodRepOrdini`) REFERENCES `reportordini`(`CodRepOrdini`)
    ON DELETE CASCADE ON UPDATE CASCADE,
  ADD FOREIGN KEY(`NomeSpecie`) REFERENCES `specie`(`Nome`)
    ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `elementinecessarispecie`
  ADD FOREIGN KEY(`NomeSpecie`) REFERENCES `specie`(`Nome`)
    ON DELETE CASCADE ON UPDATE CASCADE,
  ADD FOREIGN KEY(`NomeElemento`) REFERENCES `elemento`(`Nome`)

```

```

ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `elementiterreno`
  ADD FOREIGN KEY(`NomeElemento`) REFERENCES `elemento`(`Nome`)
  ON DELETE CASCADE ON UPDATE CASCADE,
  ADD FOREIGN KEY(`CodTerreno`) REFERENCES `terreno`(`CodTerreno`)
  ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `componentiterreno`
  ADD FOREIGN KEY(`CodTerreno`) REFERENCES `terreno`(`CodTerreno`)
  ON DELETE CASCADE ON UPDATE CASCADE,
  ADD FOREIGN KEY(`NomeComponente`) REFERENCES `componente`(`Nome`)
  ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `salute`
  ADD FOREIGN KEY(`CodContenitore`)
  REFERENCES `contenitore`(`CodContenitore`)
  ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `elementistatosalute`
  ADD FOREIGN KEY(`CodContenitore`,`Timestamp`)
  REFERENCES `salute`(`CodContenitore`,`Timestamp`)
  ON DELETE CASCADE ON UPDATE CASCADE,
  ADD FOREIGN KEY(`NomeElemento`) REFERENCES `elemento`(`Nome`)
  ON DELETE CASCADE ON UPDATE CASCADE;

-- I ripiani e i contenitori si possono spostare (ON DELETE SET NULL)
ALTER TABLE `contenitore`
  ADD FOREIGN KEY(`CodRipiano`) REFERENCES `ripiano`(`CodRipiano`)
  ON DELETE SET NULL ON UPDATE CASCADE;
ALTER TABLE `ripiano`
  ADD FOREIGN KEY(`CodSezione`) REFERENCES `sezione`(`CodSezione`)
  ON DELETE SET NULL ON UPDATE CASCADE;

-- Le sezioni e le serre no (ON DELETE CASCADE)
ALTER TABLE `sezione`
  ADD FOREIGN KEY(`CodSerra`) REFERENCES `serra`(`CodSerra`)
  ON DELETE CASCADE ON UPDATE CASCADE;
ALTER TABLE `serra`
  ADD FOREIGN KEY(`CodSede`) REFERENCES `sede`(`CodSede`)
  ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `spazio`
  ADD FOREIGN KEY(`Utente`) REFERENCES `account`(`Nickname`)
  ON DELETE SET NULL ON UPDATE CASCADE;

ALTER TABLE `verticisetto`
  ADD FOREIGN KEY(`CodSetto`) REFERENCES `setto`(`CodSetto`)
  ON DELETE CASCADE ON UPDATE CASCADE,
  ADD FOREIGN KEY(`PosX`,`PosY`) REFERENCES `vertice`(`PosX`,`PosY`)
  ON DELETE NO ACTION ON UPDATE NO ACTION;

ALTER TABLE `arredamento`
  ADD FOREIGN KEY(`CodSpazio`) REFERENCES `spazio`(`CodSpazio`)
  ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `vasiarredamento`
  ADD FOREIGN KEY(`CodVaso`) REFERENCES `vaso`(`CodVaso`)
  ON DELETE CASCADE ON UPDATE CASCADE,
  ADD FOREIGN KEY(`CodSpazio`,`Versione`)
  REFERENCES `arredamento`(`CodSpazio`,`Versione`)
  ON DELETE CASCADE ON UPDATE CASCADE,

```

```

ADD FOREIGN KEY(`CodPianta`) REFERENCES `pianta`(`CodPianta`)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `piantearredamentoinpienaterra`
ADD FOREIGN KEY(`CodSpazio`,`Versione`)
REFERENCES `arredamento`(`CodSpazio`,`Versione`)
ON DELETE CASCADE ON UPDATE CASCADE,
ADD FOREIGN KEY(`CodPianta`) REFERENCES `pianta`(`CodPianta`)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `somministrazioneconcimazione`
ADD FOREIGN KEY(`NomeElemento`) REFERENCES `elemento`(`Nome`)
ON DELETE CASCADE ON UPDATE CASCADE,
ADD FOREIGN KEY(`CodManutenzione`)
REFERENCES `manutenzione`(`CodManutenzione`)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `esigenzaconcimazionepianta`
ADD FOREIGN KEY(`CodPianta`) REFERENCES `pianta`(`CodPianta`)
ON DELETE CASCADE ON UPDATE CASCADE,
ADD FOREIGN KEY(`CodManutenzione`)
REFERENCES `manutenzione`(`CodManutenzione`)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `periodispecie`
ADD FOREIGN KEY(`NomeSpecie`) REFERENCES `specie`(`Nome`)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `reportdiagnostica`
ADD FOREIGN KEY(`CodPianta`) REFERENCES `pianta`(`CodPianta`)
ON DELETE CASCADE ON UPDATE CASCADE,
ADD FOREIGN KEY(`CodTerreno`) REFERENCES `terreno`(`CodTerreno`)
ON DELETE SET NULL ON UPDATE CASCADE;

ALTER TABLE `sintomireport`
ADD FOREIGN KEY(`CodPianta`,`Timestamp`)
REFERENCES `reportdiagnostica`(`CodPianta`,`Timestamp`)
ON DELETE CASCADE ON UPDATE CASCADE,
ADD FOREIGN KEY(`CodSintomo`) REFERENCES `sintomo`(`CodSintomo`)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `patologiereport`
ADD FOREIGN KEY(`CodPianta`,`Timestamp`)
REFERENCES `reportdiagnostica`(`CodPianta`,`Timestamp`)
ON DELETE CASCADE ON UPDATE CASCADE,
ADD FOREIGN KEY(`CodPatologia`) REFERENCES `patologia`(`CodPatologia`)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `periodiprodotti`
ADD FOREIGN KEY(`NomeProdotto`) REFERENCES `prodotto`(`Nome`)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `principiattiviprodotto`
ADD FOREIGN KEY(`NomeProdotto`) REFERENCES `prodotto`(`Nome`)
ON DELETE CASCADE ON UPDATE CASCADE,
ADD FOREIGN KEY(`NomePrincipioAttivo`)
REFERENCES `principioattivo`(`Nome`)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `prodottocombatte`
ADD FOREIGN KEY(`NomeProdotto`) REFERENCES `prodotto`(`Nome`)
ON DELETE CASCADE ON UPDATE CASCADE,

```

```

ADD FOREIGN KEY(`NomeAgentePatogeno`) REFERENCES `agentepatogeno`(`Nome`)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `agentipatogenipatologia`
ADD FOREIGN KEY(`CodPatologia`) REFERENCES `patologia`(`CodPatologia`)
ON DELETE CASCADE ON UPDATE CASCADE,
ADD FOREIGN KEY(`NomeAgentePatogeno`) REFERENCES `agentepatogeno`(`Nome`)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `prodottipatologia`
ADD FOREIGN KEY(`CodPatologia`) REFERENCES `patologia`(`CodPatologia`)
ON DELETE CASCADE ON UPDATE CASCADE,
ADD FOREIGN KEY(`NomeProdotto`) REFERENCES `prodotto`(`Nome`)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `sintomipatologia`
ADD FOREIGN KEY(`CodPatologia`) REFERENCES `patologia`(`CodPatologia`)
ON DELETE CASCADE ON UPDATE CASCADE,
ADD FOREIGN KEY(`CodSintomo`) REFERENCES `sintomo`(`CodSintomo`)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `immaginisintomi`
ADD FOREIGN KEY(`CodSintomo`) REFERENCES `sintomo`(`CodSintomo`)
ON DELETE CASCADE ON UPDATE CASCADE,
ADD FOREIGN KEY(`URL`) REFERENCES `immagine`(`URL`)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `prodottitrattamento`
ADD FOREIGN KEY(`NomeProdotto`) REFERENCES `prodotto`(`Nome`)
ON DELETE CASCADE ON UPDATE CASCADE,
ADD FOREIGN KEY(`CodManutenzione`)
REFERENCES `manutenzione`(`CodManutenzione`)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `manutenzione`
ADD FOREIGN KEY(`CodPianta`) REFERENCES `pianta`(`CodPianta`)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `periodimanutenzione`
ADD FOREIGN KEY(`CodManutenzione`)
REFERENCES `manutenzione`(`CodManutenzione`)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `reportassunzioni`
ADD FOREIGN KEY(`CodSede`) REFERENCES `sede`(`CodSede`)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `settore`
ADD FOREIGN KEY(`CodSpazio`) REFERENCES `spazio`(`CodSpazio`)
ON DELETE CASCADE ON UPDATE CASCADE;

```

```

--
-- Popolamento
--

```

```

INSERT INTO `media` VALUES
('./forum/img1.png'),
('./forum/img2.png'),
('./forum/img3.png'),
('./forum/img4.png'),
('./forum/img5.png'),
('./forum/img6.png'),
('./forum/img7.png'),

```

```
( './forum/img8.png'),
( './forum/img9.png'),
( './forum/img10.png');
```

```
INSERT INTO `thread` (`CodThread`, `Nome`) VALUES
('1', 'Bonsai Mania'),
('2', 'Foto delle vostre piante preferite!'),
('3', 'Dove posso trovare i migliori pini?'),
('4', 'Piante natalizie'),
('5', 'La vostra pianta preferita?'),
('6', 'Viti: una passione'),
('7', 'Garden designers'),
('8', 'Vasi per cactus'),
('9', 'Suggerimenti'),
('10', 'Piante tropicali');
```

```
INSERT INTO `vertice` VALUES
('1', '1'), ('1', '2'), ('1', '3'), ('1', '4'), ('1', '5'), ('1', '6'),
('1', '7'), ('1', '8'), ('1', '9'), ('1', '10'),
('2', '1'), ('2', '2'), ('2', '3'), ('2', '4'), ('2', '5'), ('2', '6'),
('2', '7'), ('2', '8'), ('2', '9'), ('2', '10'),
('3', '1'), ('3', '2'), ('3', '3'), ('3', '4'), ('3', '5'), ('3', '6'),
('3', '7'), ('3', '8'), ('3', '9'), ('3', '10'),
('4', '1'), ('4', '2'), ('4', '3'), ('4', '4'), ('4', '5'), ('4', '6'),
('4', '7'), ('4', '8'), ('4', '9'), ('4', '10'),
('5', '1'), ('5', '2'), ('5', '3'), ('5', '4'), ('5', '5'), ('5', '6'),
('5', '7'), ('5', '8'), ('5', '9'), ('5', '10'),
('6', '1'), ('6', '2'), ('6', '3'), ('6', '4'), ('6', '5'), ('6', '6'),
('6', '7'), ('6', '8'), ('6', '9'), ('6', '10'),
('7', '1'), ('7', '2'), ('7', '3'), ('7', '4'), ('7', '5'), ('7', '6'),
('7', '7'), ('7', '8'), ('7', '9'), ('7', '10'),
('8', '1'), ('8', '2'), ('8', '3'), ('8', '4'), ('8', '5'), ('8', '6'),
('8', '7'), ('8', '8'), ('8', '9'), ('8', '10'),
('9', '1'), ('9', '2'), ('9', '3'), ('9', '4'), ('9', '5'), ('9', '6'),
('9', '7'), ('9', '8'), ('9', '9'), ('9', '10'),
('10', '1'), ('10', '2'), ('10', '3'), ('10', '4'), ('10', '5'),
('10', '6'), ('10', '7'), ('10', '8'), ('10', '9'), ('10', '10');
```

```
-- Da Wikipedia: Famiglia -> Genere -> Specie
-- Esempi di specie e generi: goo.gl/KSZj8F
-- Esempi di cultivars: en.wikipedia.org/wiki/Lists\_of\_cultivars
```

```
INSERT INTO `specie` (`Nome`,
`Genere`,
`Cultivar`,
`Infestante`,
`DimMax`,
`PosLuce`,
`TipoLuce`,
`OreLuceVegetativo`,
`TempMax`,
`ConsistenzaTerreno`,
`DistanzaMinConflitto`,
`Dioica`,
`TempMin`,
`OreLuceRiposo`,
`Fogliame`,
`NumIrrigGiornaliereVegetativo`,
`NumIrrigGiornaliereRiposo`,
`QuantitaIrrigVegetativo`,
`QuantitaIrrigRiposo`,
`PeriodicitaIrrigVegetativo`,
`PeriodicitaIrrigRiposo`) VALUES
```

```

('Mays', 'Zea', 'Mais', '0', '20', 'Pieno sole', 'diretta',
 '9', '48', 'compatto', '15', '0', '1', '6', 'seghettato',
 '2', '1', '20', '10', '10', '20'),
('Aestivum', 'Triticum', 'Grano', '0', '30', 'Pieno sole', 'diretta',
 '11', '40', 'laterico', '15', '0', '5', '6', 'lacerato',
 '1', '1', '15', '10', '7', '15'),
('Minor', 'Ulmus', 'Olmo', '0', '70', 'Mezz\ombra', 'indiretta',
 '10', '45', 'argilloso', '35', '0', '10', '7', 'pennatosetto',
 '2', '0', '10', '7', '10', '30'),
('Perennis', 'Bellis', 'Pratolina', '0', '25', 'Mezz\ombra', 'diretta',
 '8', '29', 'sabbioso', '15', '1', '15', '8', 'lacerato',
 '1', '1', '20', '15', '5', '14'),
('Officinalis', 'Taraxacum', 'Tarassaco', '1', '15', 'Ombra', 'indiretta',
 '12', '37', 'sciolto', '12', '0', '5', '4', 'seghettato',
 '1', '1', '25', '10', '15', '20'),
('Alba', 'Populus', 'Gattice', '0', '30', 'Ombra', 'indiretta',
 '7', '39', 'ibrido', '15', '1', '10', '6', 'lobato',
 '1', '0', '18', '18', '20', '20'),
('Avium', 'Prunus', 'Ciliegio', '0', '90', 'Pieno sole', 'diretta',
 '11', '35', 'compatto', '60', '1', '7', '5', 'pennatifido',
 '2', '0', '13', '10', '15', '20'),
('Moraceae', 'Ficus', 'Trigmus', '0', '30', 'Mezz\ombra', 'indiretta',
 '9', '46', 'argilloso', '35', '0', '10', '7', 'pennatosetto',
 '2', '0', '10', '7', '10', '30'),
('Tiliaceae', 'Tilia', 'Triticum', '0', '25', 'Pieno sole', 'diretta',
 '8', '28', 'sabbioso', '15', '1', '15', '8', 'lacerato',
 '1', '1', '20', '15', '5', '14'),
('Rosaceae', 'Spiraea', 'Gattice', '1', '15', 'Mezz\ombra', 'indiretta',
 '11', '31', 'sciolto', '12', '0', '5', '4', 'seghettato',
 '1', '1', '25', '10', '15', '20');

```

```

INSERT INTO `account` ( `Nickname`,
                        `CittaResidenza`,
                        `Password`,
                        `Email`,
                        `Nome`,
                        `Cognome`,
                        `DomandaSegreta`,
                        `RispostaSegreta`) VALUES
('jack86', 'Londra', 'om12km3lk9', 'jk86@qwemk.com',
 'Jack', 'White', 'Nome del gatto di tua suocera', 'Fuffy'),
('John244', 'Milano', '3loi4o1k2m3l', 'hn244@ekk.com',
 'John', 'Black', 'Dove sei nato', 'Bologna'),
('RobJDK', 'Roma', 'l3ep12ok3', 'robjdk@odlk.com',
 'Rob', 'Huston', 'Cognome di tua madre', 'White'),
('MickeyMouse12', 'Napoli', 'jri3j412klm', 'mickeymouse12@ksmc.com',
 'Mickey', 'Mouse', 'Nome del cane', 'Bobby'),
('RoseLover', 'Venezia', 'kop4lpo2k4', 'roselover11@col.com',
 'Fred', 'Durst', 'Piatto preferito', 'Pizza'),
('jonny1', 'Londra', 'olij14ò2omkk', 'jonny1_ekd@ekdm.com',
 'Jonny', 'London', 'Film preferito', 'Pulp Fiction'),
('Alex_9', 'Roma', 'o3i4ulh24km', 'alex9@jdem.com',
 'Alex', 'White', 'Cantante preferito', 'Freddy Mercury'),
('Eric', 'Napoli', 'jbslkjrlkj413l4il', 'eric_lsk@edd.com',
 'Eric', 'Black', 'Data di laurea', '3 novembre 2015'),
('Paul311', 'Bologna', '5ji453j6o1', 'kelw@wol.com',
 'Paul', 'Jhonson', 'Quanto sei alto', '176'),
('PeterWhite', 'Roma', 'l2okm3n3knj2', 'sksld@amz.com',
 'Peter', 'White', 'Film preferito', 'Titanic');

```

```

INSERT INTO `spazio` VALUES
('1', 'verde', 'Eric'),

```

```

('2', 'non verde', 'jonny1'),
('3', 'non verde', 'jack86'),
('4', 'verde', 'RoseLover'),
('5', 'non verde', 'Eric'),
('6', 'verde', 'MickeyMousel2'),
('7', 'non verde', 'RobJDK'),
('8', 'verde', 'MickeyMousel2'),
('9', 'non verde', 'Eric'),
('10', 'verde', 'RobJDK');

INSERT INTO `settore` ( `CodSettore`,
                        `Tipo`,
                        `Esposizione`,
                        `NumOreLuce`,
                        `CodSpazio`) VALUES
('1', 'piena terra', 'nord', '11', '9'),
('2', 'pavimentato', 'sud-est', '8', '6'),
('3', 'piena terra', 'ovest', '5', '5'),
('4', 'pavimentato', 'sud', '0', '10'),
('5', 'pavimentato', 'nord-est', '5', '9'),
('6', 'piena terra', 'est', '12', '4'),
('7', 'piena terra', 'nord', '7', '6'),
('8', 'pavimentato', 'sud-ovest', '0', '3'),
('9', 'piena terra', 'nord', '6', '2'),
('10', 'pavimentato', 'nord-ovest', '7', '1');

INSERT INTO `verticisetto` (`CodSettore`, `PosX`, `PosY`) VALUES
('1', '1', '1'), ('1', '4', '1'), ('1', '1', '5'), ('1', '4', '5'),
('2', '1', '2'), ('2', '4', '2'), ('2', '1', '10'), ('2', '4', '10'),
('3', '3', '5'), ('3', '5', '10'), ('3', '1', '10'),
('4', '5', '1'), ('4', '2', '2'), ('4', '8', '2'), ('4', '3', '8'),
('4', '5', '8'),
('5', '7', '4'), ('5', '10', '4'), ('5', '7', '10'), ('5', '10', '10'),
('6', '1', '6'), ('6', '4', '6'), ('6', '1', '1'),
('7', '5', '1'), ('7', '5', '5'), ('7', '9', '1'),
('8', '2', '2'), ('8', '2', '7'), ('8', '7', '2'), ('8', '7', '7'),
('9', '4', '4'), ('9', '4', '8'), ('9', '8', '4'), ('9', '8', '8'),
('10', '3', '3'), ('10', '3', '6'), ('10', '6', '3'), ('10', '6', '6');

INSERT INTO `post` VALUES
('2016-12-27 11:56:08', 'jack86', '3.4', 'Secondo me...', '4'),
('2015-09-12 12:14:58', 'Paul311', '3.3', 'La pianta che...', '1'),
('2016-02-23 15:38:45', 'PeterWhite', '4.7', 'Penso che...', '10'),
('2016-01-10 09:12:34', 'Eric', '4.9', 'Secondo me...', '9'),
('2015-05-09 14:44:40', 'RoseLover', '1.4', 'A mio parere...', '5'),
('2014-12-07 18:21:17', 'Alex_9', '2.3', 'Secondo me...', '6'),
('2016-07-01 07:20:57', 'jack86', '4.1', 'Io credo che...', '3'),
('2015-08-29 11:16:49', 'RobJDK', '3.0', 'Secondo me...', '2'),
('2016-10-28 17:38:38', 'RoseLover', '2.9', 'Secondo me...', '6'),
('2015-09-30 08:45:18', 'jack86', '5.0', 'Forse potrebbe...', '4');

INSERT INTO `mediapost` VALUES
('2016-12-27 11:56:08', 'jack86', './forum/img3.png'),
('2016-02-23 15:38:45', 'PeterWhite', './forum/img4.png'),
('2016-02-23 15:38:45', 'PeterWhite', './forum/img9.png'),
('2016-02-23 15:38:45', 'PeterWhite', './forum/img2.png'),
('2016-02-23 15:38:45', 'PeterWhite', './forum/img8.png'),
('2016-01-10 09:12:34', 'Eric', './forum/img4.png'),
('2014-12-07 18:21:17', 'Alex_9', './forum/img7.png'),
('2016-07-01 07:20:57', 'jack86', './forum/img1.png'),
('2016-07-01 07:20:57', 'jack86', './forum/img6.png'),
('2016-07-01 07:20:57', 'jack86', './forum/img5.png'),

```

```

('2015-08-29 11:16:49', 'RobJDK', './forum/img1.png'),
('2015-09-30 08:45:18', 'jack86', './forum/img10.png'),
('2015-09-30 08:45:18', 'jack86', './forum/img6.png');

INSERT INTO `specieappassionatoaccount` VALUES
('Avium', 'RobJDK'),
('Rosaceae', 'jack86'),
('Officinalis', 'PeterWhite'),
('Tiliaceae', 'Alex_9'),
('Alba', 'RoseLover'),
('Moraceae', 'MickeyMouse12'),
('Minor', 'John244'),
('Avium', 'Eric'),
('Minor', 'jack86'),
('Mays', 'Alex_9');

INSERT INTO `ordine` VALUES
('1', '2016-01-14 09:30:15', 'pendente', 'Paul311'),
('2', '2016-10-16 18:43:18', 'in processazione', 'Alex_9'),
('3', '2016-02-03 14:15:46', 'spedito', 'PeterWhite'),
('4', '2016-07-19 16:08:05', 'in preparazione', 'MickeyMouse12'),
('5', '2016-09-29 05:05:00', 'in processazione', 'Alex_9'),
('6', '2016-05-05 09:13:48', 'pendente', 'jack86'),
('7', '2016-11-09 13:59:56', 'in preparazione', 'Eric'),
('8', '2016-04-15 22:48:37', 'in processazione', 'jack86'),
('9', '2016-03-23 14:36:47', 'spedito', 'Eric'),
('10', '2016-12-27 23:14:32', 'evaso', 'jack86');

INSERT INTO `sede` (`CodSede`,
`NumDipendenti`,
`Indirizzo`,
`Nome`,
`Capienza`) VALUES
('1', '35', 'Via Roma 34, Pisa, Italia', 'Da Giorgi', '400'),
('2', '12', 'Via Venezia 19, Roma, Italia', 'PuntoPianta', '300'),
('3', '50', 'Via Diotisalvi 9, Napoli, Italia', 'GarDesign', '100'),
('4', '38', 'Via Giorgi 5, Bologna, Italia', 'Magazzino Bologna', '200'),
('5', '10', 'St James Street 12, Londra, UK', 'Il Pollice Verde', '300'),
('6', '26', 'Via G. Paolo II 34, Bari, Italia', 'SOS Piante', '250'),
('7', '31', 'Via Garibaldi 34, Brindisi, Italia', 'Mania Bonsai', '120'),
('8', '42', 'Via XIV Maggio 1, Pescara, Italia', 'Da Piero', '300'),
('9', '68', 'Via Togliatti 3, Brescia, Italia', 'Da zio Mario', '250'),
('10', '18', 'Via Veneto 2, Milano, Italia', 'Tutto in Verde', '300');

INSERT INTO `serra` ( `CodSerra`,
`Nome`,
`CodSede`,
`Indirizzo`,
`Larghezza`,
`Altezza`,
`Capienza`) VALUES
('1', 'Serral-1', '1', 'Via Roma 18, Pisa, Italia', '50', '50', '100'),
('2', 'Serra2-1', '1', 'Via Roma 19, Pisa, Italia', '50', '50', '100'),
('3', 'Serra3-1', '1', 'Via Roma 20, Pisa, Italia', '50', '50', '100'),
('4', 'Serra4-1', '1', 'Via Roma 21, Pisa, Italia', '50', '50', '100'),
('5', 'Serral-2', '2', 'Via Vittorio Veneto 26, Venezia, Italia', '70',
'60', '150'),
('6', 'Serra2-2', '2', 'Via Vittorio Veneto 27, Venezia, Italia', '70',
'60', '150'),
('7', 'Serral-3', '3', 'Via Togliatti 7, Bari, Italia', '40', '40', '80'),
('8', 'Serral-5', '5', 'Via Giovanni Paolo II 14, Milano, Italia', '80',
'50', '90'),

```



```

('9', 'Serra2-5', '5', 'Via Giovanni Paolo II 15, Milano, Italia', '80',
'50', '90'),
('10', 'Serral-6', '6', 'Via Bovio 2, Ancona, Italia', '100', '50', '120'),
('11', 'Serra2-6', '6', 'Via Bovio 3, Ancona, Italia', '100', '50', '120'),
('12', 'Serral-9', '9', 'Via Garibaldi 9, Catania, Italia', '100', '120',
'180'),
('13', 'Serral-10', '10', 'Via XIV Maggio 10, Milano, Italia', '40', '50',
'100');

```

```

INSERT INTO `sezione` ( `CodSezione`,
`Nome`,
`CodSerra`,
`Capienza`,
`Irrigazione`,
`Illuminazione`,
`Umidita`,
`Temperatura`,
`Quarantena`) VALUES
('1', 'Sezionel-1-1', '1', '20', '75', '50', '13', '20', '0'),
('2', 'Sezione2-1-1', '1', '30', '71', '67', '9', '21', '0'),
('3', 'Sezionel-2-1', '2', '25', '64', '55', '10', '20.5', '0'),
('4', 'Sezionel-3-1', '3', '35', '67', '53', '12', '21.5', '0'),
('5', 'Sezionel-4-1', '4', '20', '68', '51', '9.5', '19', '0'),
('6', 'Sezionel-1-2', '5', '40', '68', '51', '9.5', '19', '0'),
('7', 'Sezionel-2-2', '6', '45', '76', '64', '5', '22', '0'),
('8', 'Sezionel-1-3', '7', '25', '84', '71', '11', '22.5', '0'),
('9', 'Sezione2-1-3', '7', '10', '70', '35', '18', '18', '1'),
('10', 'Sezionel-1-6', '10', '55', '66', '68', '12', '20.5', '0');

```

```

INSERT INTO `ripiano` ( `CodRipiano`,
`CodSezione`,
`Capienza`,
`Irrigazione`) VALUES
('1', '1', '5', '71'),
('2', '1', '5', '72'),
('3', '1', '5', '70'),
('4', '1', '5', '66'),
('5', '2', '10', '69'),
('6', '2', '15', '68'),
('7', '3', '10', '65'),
('8', '3', '15', '65'),
('9', '4', '20', '71'),
('10', '4', '15', '74'),
('11', '6', '20', '79'),
('12', '6', '20', '64'),
('13', '8', '20', '68'),
('14', '9', '10', '77'),
('15', '10', '40', '75');

```

```

INSERT INTO `contenitore` ( `CodContenitore`,
`Dimensione`,
`Idratazione`,
`Irrigazione`,
`CodRipiano`) VALUES
('1', '50', '72', '81.5', '1'),
('2', '30', '71', '78', '2'),
('3', '40', '59.5', '79', '4'),
('4', '55', '69', '82', '5'),
('5', '20', '74.5', '84', '7'),
('6', '70', '78', '87.5', '8'),
('7', '10', '82', '77', '9'),
('8', '30', '58.5', '84', '10'),

```

```

('9', '35', '71', '79', '11'),
('10', '30', '68', '85.5', '12'),
('11', '50', '64.5', '80', '13'),
('12', '45', '74', '83.5', '15');

INSERT INTO `salute` VALUES
('1', '2014-04-20 09:30:15', '15', '40.5', '23'),
('1', '2014-05-20 11:09:47', '16', '37', '27.5'),
('1', '2014-08-20 13:41:17', '15.5', '38', '24'),
('3', '2015-10-14 17:13:02', '14.7', '45.1', '29'),
('3', '2015-11-14 14:16:57', '17.8', '40', '28.5'),
('5', '2015-05-08 05:06:12', '15', '38', '26'),
('5', '2015-06-08 06:11:35', '15.6', '38', '26.2'),
('5', '2015-07-08 05:07:43', '17', '37', '26.9'),
('5', '2015-08-08 07:01:55', '16.1', '38', '27.4'),
('6', '2016-08-19 04:01:14', '19', '42.3', '25'),
('6', '2015-10-19 03:02:12', '18.6', '9', '26');

INSERT INTO `elemento` VALUES
('calcio', '16'),
('magnesio', '20.8'),
('ferro', '25'),
('potassio', '11'),
('azoto', '12'),
('fosforo', '13.7'),
('zolfo', '9'),
('boro', '19'),
('manganese', '7'),
('rame', '23'),
('zinco', '8'),
('molibdeno', '10.5'),
('cloro', '15'),
('nichel', '11.3');

INSERT INTO `elementistatosalute` VALUES
('1', '2014-04-20 09:30:15', 'calcio', '16.2'),
('1', '2014-05-20 11:09:47', 'azoto', '13.6'),
('1', '2014-05-20 11:09:47', 'boro', '41.0'),
('3', '2015-10-14 17:13:02', 'zinco', '21.5'),
('3', '2015-10-14 17:13:02', 'ferro', '9.3'),
('5', '2015-05-08 05:06:12', 'potassio', '11.5'),
('5', '2015-05-08 05:06:12', 'manganese', '4.6'),
('5', '2015-06-08 06:11:35', 'nichel', '24.9'),
('5', '2015-07-08 05:07:43', 'azoto', '15.4'),
('6', '2016-08-19 04:01:14', 'zolfo', '7.5'),
('6', '2016-08-19 04:01:14', 'molibdeno', '5.1'),
('6', '2015-10-19 03:02:12', 'cloro', '15.2');

-- it.wikipedia.org/wiki/Reazione_del_terreno#Classificazione_dei_terreni
-- PH: acido [3.5, 6.9] | alcalino [7.0, 9.0]
-- Permeabilita: [0.0001, 10^8]
INSERT INTO `terreno` VALUES
('1', '6.2', 'scioltto', '204.482'),
('2', '3.9', 'argilloso', '204.482'),
('3', '4.2', 'sabbioso', '204.482'),
('4', '6.0', 'laterico', '204.482'),
('5', '6.8', 'compatto', '204.482'),
('6', '3.7', 'argilloso', '204.482'),
('7', '4.2', 'compatto', '204.482'),
('8', '5.3', 'scioltto', '204.482'),
('9', '4.7', 'ibrido', '204.482'),
('10', '5.9', 'laterico', '204.482');

```

```

INSERT INTO `componente` VALUES
  ('sabbia'), ('torba'), ('ghiaia'), ('argilla'), ('limo');

INSERT INTO `componentiterreno` VALUES
  ('1', 'sabbia', '39'), ('1', 'torba', '8'), ('1', 'argilla', '41'),
  ('2', 'limo', '41'), ('2', 'torba', '13'), ('2', 'sabbia', '11'),
  ('3', 'ghiaia', '12'), ('3', 'argilla', '74.9'),
  ('4', 'sabbia', '75'), ('4', 'limo', '19'), ('4', 'torba', '5'),
  ('5', 'limo', '12.2'), ('5', 'ghiaia', '48.5'), ('5', 'sabbia', '18'),
  ('6', 'torba', '24'), ('6', 'sabbia', '14'), ('6', 'argilla', '24'),
  ('7', 'sabbia', '42'), ('7', 'torba', '45'),
  ('8', 'argilla', '49'), ('8', 'limo', '17'), ('8', 'sabbia', '18.5'),
  ('9', 'ghiaia', '71'), ('9', 'torba', '14'),
  ('10', 'argilla', '23.5'), ('10', 'ghiaia', '48'), ('10', 'sabbia', '24');

INSERT INTO `elementiterreno` ( `CodTerreno`,
                                `NomeElemento`,
                                `Concentrazione`) VALUES
  ('1', 'calcio', '14.2'), ('1', 'rame', '18'), ('1', 'ferro', '62'),
  ('2', 'zinco', '82'), ('2', 'cloro', '12.4'),
  ('3', 'nichel', '46.1'), ('3', 'fosforo', '41'), ('3', 'azoto', '3.9'),
  ('4', 'ferro', '41.6'), ('4', 'boro', '16.5'), ('4', 'azoto', '37'),
  ('5', 'molibdeno', '18'), ('5', 'cloro', '42'), ('5', 'boro', '7.8'),
  ('6', 'boro', '71'), ('6', 'azoto', '4.1'),
  ('7', 'zolfo', '47'), ('7', 'cloro', '13'), ('7', 'fosforo', '31.4'),
  ('8', 'cloro', '70'), ('8', 'potassio', '12.5'), ('8', 'boro', '6.3'),
  ('9', 'zinco', '19.7'), ('9', 'azoto', '76.1'),
  ('10', 'magnesio', '56'), ('10', 'molibdeno', '14.3'),
  ('10', 'cloro', '4');

INSERT INTO `elementinecessarispecie` VALUES
  ('Mays', 'calcio', '13.4'), ('Mays', 'azoto', '41.2'),
  ('Mays', 'fosforo', '36.2'), ('Mays', 'cloro', '7.8'),
  ('Aestivum', 'ferro', '7.9'), ('Aestivum', 'zolfo', '28.4'),
  ('Aestivum', 'rame', '15.6'),
  ('Minor', 'magnesio', '9.4'), ('Minor', 'azoto', '18.6'),
  ('Perennis', 'potassio', '47.1'), ('Perennis', 'fosforo', '11.9'),
  ('Perennis', 'zolfo', '19.5'), ('Perennis', 'molibdeno', '3.5'),
  ('Officinalis', 'azoto', '29.4'), ('Officinalis', 'zinco', '2.9'),
  ('Alba', 'zinco', '52.3'),
  ('Avium', 'molibdeno', '16.8'), ('Avium', 'zolfo', '31.2'),
  ('Moraceae', 'ferro', '34.2'), ('Moraceae', 'rame', '12.2'),
  ('Moraceae', 'cloro', '10.2'),
  ('Tiliaceae', 'fosforo', '15.3'), ('Tiliaceae', 'nichel', '9.0'),
  ('Rosaceae', 'calcio', '18.5'), ('Rosaceae', 'boro', '51.2'),
  ('Rosaceae', 'manganese', '5.6');

INSERT INTO `pianta` ( `CodPianta`,
                       `NomeSpecie`,
                       `DimAttuale`,
                       `Prezzo`,
                       `CodTerreno`,
                       `CodContenitore`,
                       `CodOrdine`) VALUES
  ('1', 'Mays', '28', '15', '4', NULL, '10'),
  ('2', 'Mays', '23', '20', '4', '8', NULL),
  ('3', 'Aestivum', '12', '48', '2', '9', '6'),
  ('4', 'Minor', '10', '10', '1', NULL, '3'),
  ('5', 'Perennis', '70', '60', '6', '6', NULL),
  ('6', 'Officinalis', '40', '30', '2', NULL, '9'),
  ('7', 'Avium', '47', '20', '8', '1', NULL),

```

```

('8', 'Moraceae', '15', '30', '7', '5', '1'),
('9', 'Tiliaceae', '19', '13', '10', '11', NULL),
('10', 'Rosaceae', '5', '8', '9', '7', '2'),
('11', 'Alba', '41', '40', '5', '12', '8'),
('12', 'Perennis', '34', '38', '3', '3', '5'),
('13', 'Minor', '29', '20', '2', NULL, '7'),
('14', 'Moraceae', '18', '15', '10', NULL, '4');

INSERT INTO `manutenzione` (`CodManutenzione`,
                           `CodPianta`,
                           `TipoManutenzione`, /* potatura, rinvaso,
                                                concimazione, trattamento */
                           `TipoCreazione`, /* su richiesta,
                                                programmata, automatica */
                           `Costo`, -- euro
                           -- it.wikipedia.org/wiki/Potatura#Metodi_di_potatura
                           `TipoPotatura`,
                           `TipoSomm`, -- disciolto, nebulizzato
                           `DataInizio`,
                           `NumIntervAnnuali`,
                           `Scadenza`) VALUES
('1', '1', 'potatura', 'su richiesta', '25', 'capitozzatura', NULL,
 '2013-05-02', '3', '2020-04-05'),
('2', '2', 'rinvaso', 'su richiesta', '5', NULL, NULL,
 '2013-10-05', '1', '2013-10-05'),
('3', NULL, 'concimazione', 'programmata', '7', NULL, 'disciolto',
 '2014-01-14', '12', '2018-01-14'),
('4', '4', 'trattamento', 'su richiesta', '19', NULL, 'nebulizzato',
 '2012-07-20', '5', '2012-11-20'),
('5', '5', 'potatura', 'programmata', '30', 'sfogliatura', NULL,
 '2015-10-10', '3', '2017-10-10'),
('6', '6', 'rinvaso', 'su richiesta', '17', NULL, NULL,
 '2014-05-05', '1', '2014-05-05'),
('7', '8', 'potatura', 'su richiesta', '36', 'piegatura', NULL,
 '2014-07-12', '3', '2014-9-12'),
('8', '10', 'trattamento', 'programmata', '45', NULL, 'nebulizzato',
 '2016-12-25', '12', '2018-12-25'),
('9', NULL, 'concimazione', 'programmata', '16', NULL, 'disciolto',
 '2017-01-01', '6', '2017-12-31'),
('10', '13', 'rinvaso', 'su richiesta', '8', NULL, NULL,
 '2015-10-09', '1', '2015-10-09');

INSERT INTO `prodotto` VALUES
('Muflix', 'MedProducts', '0', 'disciolto'),
('Frenox', 'GreenFix', '2', 'nebulizzato'),
('Muginex', 'MedProducts', '0', 'disciolto'),
('Trenofis', 'Plants Boss', '0', 'disciolto'),
('Azerol', 'GreenFix', '5', 'nebulizzato'),
('Picrifon', 'MedProducts', '0', 'disciolto'),
('Sorivan', 'GreenFix', '15', 'nebulizzato'),
('Cifitox', 'MedProducts', '1', 'disciolto'),
('Polinaf', 'Plants Boss', '3', 'nebulizzato'),
('Sfrefx', 'MedProducts', '0', 'nebulizzato');

INSERT INTO `prodottitrattamento` ( `CodManutenzione`,
                                    `NomeProdotto`,
                                    `Dose`) VALUES
('4', 'Muflix', '15'),
('4', 'Frenox', '21'),
('4', 'Muginex', '2'),
('4', 'Trenofis', '4'),
('4', 'Azerol', '1'),

```

```

('8', 'Picrifon', '41'),
('8', 'Sorivan', '12'),
('8', 'Cifitox', '11'),
('8', 'Polinaf', '8'),
('8', 'Sfrex', '2');

```

```

INSERT INTO `periodimanutenzione` VALUES
('1', '2000-02-01', '2000-06-01'),
('4', '2000-01-01', '2000-03-01'),
('4', '2000-04-01', '2000-06-01'),
('4', '2000-07-01', '2000-12-01'),
('5', '2000-02-01', '2000-03-01'),
('5', '2000-07-01', '2000-09-01'),
('5', '2000-10-01', '2000-11-01'),
('7', '2000-06-01', '2000-12-01'),
('8', '2000-10-01', '2001-05-01'),
('9', '2000-12-01', '2000-03-01'),
('9', '2000-08-01', '2000-09-01');

```

```

INSERT INTO `somministrazioneconcimazione` (
    `CodManutenzione`,
    `NomeElemento`,
    `Iterazione`,
    `Quantita`) VALUES
('3', 'fosforo', '1', '24.3'),
('3', 'fosforo', '2', '20.3'),
('3', 'fosforo', '3', '16.3'),
('3', 'manganese', '1', '5.0'),
('3', 'manganese', '2', '10.0'),
('9', 'cloro', '1', '13.2'),
('9', 'cloro', '2', '15.2'),
('9', 'ferro', '1', '5.4'),
('9', 'ferro', '2', '6.3'),
('9', 'ferro', '3', '7.2');

```

```

INSERT INTO `esigenzaconcimazionepianta` VALUES
('1', '3'), ('2', '3'), ('4', '3'), ('7', '3'), ('9', '3'),
('2', '9'), ('3', '9'), ('4', '9'), ('5', '9'), ('10', '9');

```

```

INSERT INTO `vaso` (`CodVaso`,
    `Materiale`,
    `DimensioneX`,
    `DimensioneY`) VALUES
('1', 'cotto', '20', '20'), --
('2', 'ceramica', '30', '30'),
('3', 'terracotta', '45', '25'),
('4', 'cemento', '70', '70'), --
('5', 'cotto', '25', '30'),
('6', 'pietra', '35', '35'), --
('7', 'ceramia', '40', '20'),
('8', 'plastica', '80', '70'), --
('9', 'terracotta', '40', '25'),
('10', 'pietra', '15', '15'); --

```

```

INSERT INTO `scheda` (
    `CodScheda`,
    `CodPianta`,
    `CodVaso`,
    `Utente`,
    `DataAcquisto`,
    `Settore`,
    `Collocazione`,
    `PosX`,
    `PosY`) VALUES

```

```

('1', '1', '6', 'jack86', '2016-06-05', '1', 'vaso', '3', '4'),
('2', '3', NULL, 'Paul311', '2016-07-12', '2', 'pienat erra', '1', '9'),
('3', '4', '1', 'RobJDK', '2016-02-13', '3', 'vaso', '3', '8'),
('4', '6', '10', 'Eric', '2016-10-09', '4', 'vaso', '4', '3'),
('5', '8', NULL, 'Eric', '2016-04-30', '5', 'piena terra', '8', '5'),
('6', '10', NULL, 'Eric', '2016-12-23', '6', 'piena terra', '1', '5'),
('7', '11', '8', 'Eric', '2016-08-04', '7', 'vaso', '6', '2'),
('8', '12', '4', 'MickeyMouse12', '2016-09-10', '8', 'vaso', '5', '3'),
('9', '13', NULL, 'RoseLover', '2016-11-18', '9',
'piena terra', '7', '6'),
('10', '14', NULL, 'Eric', '2016-01-28', '10', 'piena terra', '4', '5');

```

```

INSERT INTO `arredamento` (`CodSpazio`, `Versione`) VALUES
('1', '1'), ('1', '2'),
('2', '1'),
('3', '1'), ('3', '2'),
('4', '1'),
('5', '1'),
('6', '1'),
('7', '1'), ('7', '2'), ('7', '3'),
('8', '1'),
('9', '1'), ('9', '2'),
('10', '1');

```

```

INSERT INTO `vasiarredamento` ( `CodVaso`,
`CodSpazio`,
`Versione`,
`CodPianta`,
`PosX`,
`PosY`) VALUES
('4', '6', '1', '12', '1', '9'),
('1', '10', '1', '4', '4', '3'),
('8', '9', '1', '11', '7', '5'),
('8', '9', '2', '11', '8', '5'),
('6', '3', '1', '1', '2', '2'),
('6', '3', '2', '1', '5', '3'),
('10', '1', '1', '6', '6', '4'),
('10', '1', '2', '6', '4', '5');

```

```

INSERT INTO `piantearredamentoinpienaterra` ( `CodPianta`,
`CodSpazio`,
`Versione`,
`PosX`,
`PosY`) VALUES
('14', '9', '1', '3', '2'),
('14', '9', '2', '3', '4'),
('10', '5', '1', '3', '8'),
('13', '4', '1', '1', '5'),
('3', '6', '1', '6', '2'),
('8', '1', '1', '6', '4'),
('8', '1', '2', '4', '5');

```

```

INSERT INTO `periodispecie` ( `NomeSpecie`,
`DataInizio`,
`DataFine`,
`Tipo`) VALUES
('Mays', '2000-10-01', '2001-02-01', 'riposo'),
('Mays', '2001-02-01', '2001-08-01', 'fruttificazione'),
('Mays', '2001-08-01', '2001-10-01', 'fioritura'),
('Aestivum', '2000-01-01', '2000-06-01', 'fioritura'),
('Aestivum', '2000-06-01', '2001-01-01', 'riposo'),
('Minor', '2000-04-01', '2000-12-01', 'fruttificazione'),

```

```

('Minor', '2000-12-01', '2001-04-01', 'riposo'),
('Perennis', '2000-07-01', '2000-11-01', 'fioritura'),
('Perennis', '2000-11-01', '2001-07-01', 'riposo'),
('Officinalis', '2000-04-01', '2000-08-01', 'fruttificazione'),
('Officinalis', '2000-08-01', '2001-04-01', 'riposo'),
('Alba', '2000-09-01', '2001-03-01', 'riposo'),
('Alba', '2001-03-01', '2001-09-01', 'fioritura'),
('Avium', '2000-10-01', '2001-02-01', 'riposo'),
('Avium', '2001-02-01', '2001-10-01', 'fruttificazione'),
('Moraceae', '2000-08-01', '2000-09-01', 'fruttificazione'),
('Moraceae', '2000-09-01', '2001-08-01', 'riposo'),
('Tiliaceae', '2000-05-01', '2000-08-01', 'fruttificazione'),
('Tiliaceae', '2000-08-01', '2001-05-01', 'riposo'),
('Rosaceae', '2000-04-01', '2000-10-01', 'fioritura'),
('Rosaceae', '2000-10-01', '2001-04-01', 'riposo');

```

```

INSERT INTO `periodiprodotti` VALUES
('Muflix', '2000-02-01', '2000-12-01'),
('Frenox', '2000-01-01', '2000-11-01'),
('Muginex', '2000-02-01', '2000-05-01'),
('Muginex', '2000-06-01', '2000-11-01'),
('Trenofis', '2000-01-01', '2000-10-01'),
('Axeol', '2000-05-01', '2001-04-01'),
('Sorivan', '2000-03-01', '2000-11-01'),
('Polinaf', '2000-10-01', '2001-09-01'),
('Sfrex', '2000-07-01', '2001-01-01'),
('Sfrex', '2000-02-01', '2000-06-01');

```

```

INSERT INTO `agentepatogeno` VALUES
('Metcalfa', 'insetto'),
('Psilla', 'insetto'),
('Cicalina', 'insetto'),
('Dorifora', 'insetto'),
('Acarus', 'acaro'),
('Crittogamis', 'crittogame'),
('Fongomus', 'fungo'),
('Brucialis', 'virus'),
('Destroplatus', 'virus'),
('Platterium', 'batterio');

```

```

INSERT INTO `prodottocombatte` ( `NomeProdotto`,
                                `NomeAgentePatogeno`,
                                `Dosaggio`) VALUES
('Muflix', 'Metcalfa', '15.4'),
('Frenox', 'Psilla', '9.2'),
('Muginex', 'Cicalina', '19.2'),
('Trenofis', 'Dorifora', '17.0'),
('Axeol', 'Acarus', '9.1'),
('Picrifon', 'Crittogamis', '24.3'),
('Sorivan', 'Fongomus', '17.5'),
('Cifitox', 'Brucialis', '11.1'),
('Polinaf', 'Destroplatus', '25.7'),
('Sfrex', 'Platterium', '16.0');

```

```

INSERT INTO `patologia` VALUES
('1', '2000-01-01', '2000-12-31', '21.3', '3.5'),
('2', '2000-02-01', '2000-12-31', '4.6', '8.2'),
('3', '2000-01-01', '2000-11-30', '19.2', '2.4'),
('4', '2000-02-01', '2000-12-31', '11.2', '1.3'),
('5', '2000-01-01', '2000-11-30', '4.1', '6.9'),
('6', '2000-02-01', '2000-12-31', '9.4', '4.0'),
('7', '2000-01-01', '2000-12-31', '15.1', '3.1'),

```

```

('8', '2000-02-01', '2000-12-31', '12.7', '2.4'),
('9', '2000-01-01', '2000-12-31', '45.3', '0.8'),
('10', '2000-04-01', '2000-12-31', '19.2', '2.4');

```

```

INSERT INTO `agentipatogenipatologia` VALUES

```

```

('1', 'Metcalfa'),
('1', 'Psilla'),
('2', 'Cicalina'),
('3', 'Dorifora'),
('4', 'Acarus'),
('5', 'Crittogamis'),
('6', 'Fongomus'),
('6', 'Brucialis'),
('7', 'Fongomus'),
('8', 'Destroplantus'),
('9', 'Platterium'),
('9', 'Psilla'),
('9', 'Cicalina'),
('10', 'Acarus');

```

```

INSERT INTO `principioattivo` VALUES

```

```

('acido acetilsalicilico'),
('alcaloide'),
('morfina'),
('nicotina'),
('terpene'),
('carotene'),
('glicoside'),
('digossina'),
('atracurio'),
('aloe');

```

```

INSERT INTO `principiattiviprodotto` VALUES

```

```

('Muflix', 'acido acetilsalicilico', '15.2'),
('Muflix', 'alcaloide', '9.2'),
('Frenox', 'morfina', '16.3'),
('Muginex', 'nicotina', '11.9'),
('Trenofis', 'terpene', '25.0'),
('Trenofis', 'carotene', '7.3'),
('Axeol', 'glicoside', '17.2'),
('Picrifon', 'digossina', '28.4'),
('Sorivan', 'atracurio', '12.3'),
('Sorivan', 'aloe', '4.5'),
('Cifitox', 'nicotina', '19.2'),
('Polinaf', 'morfina', '12.1'),
('Polinaf', 'glicoside', '10.2'),
('Sfrefx', 'acido acetilsalicilico', '41.2'),
('Sfrefx', 'alcaloide', '11.4'),
('Sfrefx', 'atracurio', '2.4');

```

```

INSERT INTO `prodottipatologia` VALUES

```

```

('1', 'Muflix'),
('2', 'Frenox'),
('3', 'Muginex'),
('4', 'Trenofis'),
('5', 'Axeol'),
('6', 'Picrifon'),
('7', 'Sorivan'),
('8', 'Cifitox'),
('9', 'Polinaf'),
('10', 'Sfrefx');

```



```

INSERT INTO `sintomo` VALUES
  ('1', 'Caduta delle foglie.'),
  ('2', 'Perdita di colore delle foglie.'),
  ('3', 'Lacerazione tronco.'),
  ('4', 'Lacerazione foglie.'),
  ('5', 'Marciume.'),
  ('6', 'Appassimento.'),
  ('7', 'Ingiallimento delle foglie.'),
  ('8', 'Presenza di sali bianchi sul terriccio.'),
  ('9', 'Danni alle radici.'),
  ('10', 'Copertura della foglia di un reticolato di strisce scure.');
```

```

INSERT INTO `sintomipatologia` (`CodPatologia`, `CodSintomo`) VALUES
  ('1', '1'),
  ('1', '5'),
  ('2', '8'),
  ('2', '9'),
  ('3', '3'),
  ('4', '4'),
  ('5', '5'),
  ('5', '8'),
  ('6', '5'),
  ('6', '6'),
  ('7', '7'),
  ('8', '5'),
  ('8', '8'),
  ('9', '9'),
  ('10', '10');
```

```

INSERT INTO `immagine` VALUES
  ('./sintomi/img1.png'),
  ('./sintomi/img2.png'),
  ('./sintomi/img3.png'),
  ('./sintomi/img4.png'),
  ('./sintomi/img5.png'),
  ('./sintomi/img6.png'),
  ('./sintomi/img7.png'),
  ('./sintomi/img8.png'),
  ('./sintomi/img9.png'),
  ('./sintomi/img10.png');
```

```

INSERT INTO `immaginisintomi` (`CodSintomo`, `URL`) VALUES
  ('1', './sintomi/img1.png'),
  ('2', './sintomi/img2.png'),
  ('3', './sintomi/img3.png'),
  ('4', './sintomi/img4.png'),
  ('5', './sintomi/img5.png'),
  ('6', './sintomi/img6.png'),
  ('7', './sintomi/img7.png'),
  ('8', './sintomi/img8.png'),
  ('9', './sintomi/img9.png'),
  ('10', './sintomi/img10.png');
```

```

INSERT INTO `reportdiagnostica` (`CodPianta`, `Timestamp`) VALUES
  ('1', '2016-01-14 11:40:15'),
  ('2', '2014-10-16 06:43:18'),
  ('4', '2016-02-03 19:15:46'),
  ('5', '2014-07-19 15:06:26'),
  ('6', '2016-11-29 05:45:00'),
  ('8', '2016-05-05 11:12:41'),
  ('9', '2015-11-08 13:59:56'),
  ('11', '2015-10-12 21:48:37'),
```

```

('11', '2016-02-23 08:36:57'),
('13', '2014-11-27 22:14:32');

INSERT INTO `patologiereport` ( `CodPianta`,
                                `Timestamp`,
                                `CodPatologia`) VALUES
('1', '2016-01-14 11:40:15', '1'),
('1', '2016-01-14 11:40:15', '4'),
('2', '2014-10-16 06:43:18', '2'),
('4', '2016-02-03 19:15:46', '3'),
('5', '2014-07-19 15:06:26', '4'),
('6', '2016-11-29 05:45:00', '5'),
('6', '2016-11-29 05:45:00', '9'),
('8', '2016-05-05 11:12:41', '3'),
('8', '2016-05-05 11:12:41', '6'),
('9', '2015-11-08 13:59:56', '7'),
('11', '2015-10-12 21:48:37', '2'),
('11', '2015-10-12 21:48:37', '8'),
('11', '2016-02-23 08:36:57', '9'),
('13', '2014-11-27 22:14:32', '10');

INSERT INTO `sintomireport` (`CodPianta`, `Timestamp`, `CodSintomo`) VALUES
('1', '2016-01-14 11:40:15', '3'),
('2', '2014-10-16 06:43:18', '4'),
('4', '2016-02-03 19:15:46', '7'),
('4', '2016-02-03 19:15:46', '9'),
('5', '2014-07-19 15:06:26', '1'),
('5', '2014-07-19 15:06:26', '10'),
('5', '2014-07-19 15:06:26', '9'),
('6', '2016-11-29 05:45:00', '2'),
('8', '2016-05-05 11:12:41', '2'),
('9', '2015-11-08 13:59:56', '4'),
('9', '2015-11-08 13:59:56', '6'),
('11', '2015-10-12 21:48:37', '5'),
('11', '2016-02-23 08:36:57', '7'),
('13', '2014-11-27 22:14:32', '4'),
('13', '2014-11-27 22:14:32', '8');

INSERT INTO `reportassunzioni` VALUES
('1', '1', '2', '0', '2000-02-01', '2000-03-01'),
('2', '1', '2', '0', '2000-03-01', '2000-04-01'),
('3', '1', '2', '0', '2000-04-01', '2000-05-01'),
('4', '1', '2', '0', '2000-05-01', '2000-06-01'),
('5', '1', '2', '0', '2000-06-01', '2000-07-01'),
('6', '1', '2', '1', '2000-07-01', '2001-07-01'),
('7', '1', '2', '0', '2000-11-01', '2001-12-01'),
('8', '6', '1', '0', '2001-03-01', '2001-04-01'),
('9', '6', '1', '0', '2001-04-01', '2001-05-01'),
('10', '6', '1', '0', '2001-05-01', '2001-06-01');

INSERT INTO `reportmanutenzione` VALUES
('potatura', '1', 'Mays', '20'),
('rinvaso', '1', 'Mays', '40'),
('potatura', '1', 'Aestivum', '45'),
('potatura', '2', 'Mays', '50'),
('trattamento', '1', 'Mays', '40'),
('rinvaso', '1', 'Minor', '15'),
('potatura', '3', 'Mays', '70'),
('potatura', '1', 'Officinalis', '35'),
('trattamento', '1', 'Rosaceae', '28'),
('potatura', '2', 'Officinalis', '75');

```

```

INSERT INTO `reportordini` (`CodRepOrdini`, `DaOrdinare`, `Clima`) VALUES
  ('1', '1', 'estivo'),
  ('2', '1', 'invernale'),
  ('3', '1', 'estivo'),
  ('4', '1', 'invernale'),
  ('5', '1', 'estivo'),
  ('6', '1', 'invernale'),
  ('7', '1', 'estivo'),
  ('8', '1', 'invernale'),
  ('9', '1', 'estivo'),
  ('10', '1', 'invernale');

```

```

INSERT INTO `speciereportordini` ( `CodRepOrdini`,
                                   `NomeSpecie`,
                                   `Quantita`) VALUES
  ('1', 'Mays', '2'), ('1', 'Aestivum', '1'),
  ('3', 'Minor', '1'),
  ('4', 'Perennis', '1'),
  ('6', 'Officinalis', '1'), ('6', 'Alba', '1'),
  ('8', 'Avium', '1'),
  ('9', 'Moraceae', '1'), ('9', 'Rosaceae', '3'),
  ('10', 'Tiliaceae', '1'), ('10', 'Moraceae', '1');

```

```

--
-- Operazione 1: Ottenere il numero di posts pubblicati da un account
--

```

```

DROP PROCEDURE IF EXISTS OttieniNumPostPubblicati;
DELIMITER $$
CREATE PROCEDURE OttieniNumPostPubblicati( IN nick char(50),
                                           OUT numeropost int(11))
BEGIN
  SELECT `NumPostPubblicati` INTO numeropost
  FROM `account`
  WHERE `Nickname` = nick;
END $$
DELIMITER ;

```

```

--
-- Operazione 2: Ottenere il costo complessivo di
--               manutenzione di una specie di pianta
--

```

```

DROP PROCEDURE IF EXISTS OttieniCostoTotManutenzione;
DELIMITER $$
CREATE PROCEDURE OttieniCostoTotManutenzione(IN nomespecie char(50),
                                           OUT costo int(11))
BEGIN
  SELECT `CostoTotManutenzione` INTO costo
  FROM `specie`
  WHERE `Nome` = nomespecie;
END $$
DELIMITER ;

```

```

--
-- Operazione 3: Ottenere il numero di piante appartenenti ad una sezione
--

```

```

DROP PROCEDURE IF EXISTS OttieniNumeroPianteSezione;
DELIMITER $$
CREATE PROCEDURE OttieniNumeroPianteSezione(IN sezione int(11),
                                           OUT numeropiante int(11))
BEGIN
  SELECT `PiantePresenti` INTO numeropiante
  FROM `sezione`

```

```

        WHERE `CodSezione` = sezione;
END $$
DELIMITER ;

--
-- Operazione 4: Ottenere la specie di pianta più venduta
--
DROP PROCEDURE IF EXISTS OttieniSpeciePiuVenduta;
DELIMITER $$
CREATE PROCEDURE OttieniSpeciePiuVenduta()
BEGIN
    CREATE OR REPLACE VIEW `PiantePiuVendute` AS
        SELECT `Nome` AS `SpeciePiuVendute`
        FROM `specie`
        WHERE `NumPianteVendute` = (SELECT MAX(`NumPianteVendute`)
                                   FROM `specie`);

    SELECT *
    FROM `PiantePiuVendute`;
END $$
DELIMITER ;

--
-- Operazione 5: Inserire di una nuova pianta nel magazzino
--
/* Consiste nell'inserimento di una tupla nella tabella
`pianta` ed eventuali altre tuple relative nelle
tabelle `periodipianta`, `esigenzaconcimazionepianta`
e `elementinecessaripianta`. */

--
-- Operazione 6: Creare un nuovo arredamento
--
/* Consiste nell'inserimento di una tupla nella tabella
`arredamento` ed eventuali altre tuple relative nelle
tabelle `piantearredamentoinpienaterra` e
`vasiarredamento`. */

--
-- Operazione 7: Trovare il thread con più contenuti multimediali
--
DROP PROCEDURE IF EXISTS OttieniThreadsConPiuMedia;
DELIMITER $$
CREATE PROCEDURE OttieniThreadsConPiuMedia()
BEGIN
    CREATE OR REPLACE VIEW `ThreadsConPiuMedia` AS
        SELECT `CodThread` AS `ThreadConPiuContenutiMultimediali`
        FROM `thread`
        WHERE `NumMedia` = (SELECT MAX(`NumMedia`)
                             FROM `thread`);

    SELECT *
    FROM `ThreadsConPiuMedia`;
END $$
DELIMITER ;

--
-- Operazione 8: Ottenere la specie pianta che si ammala più spesso
--
DROP PROCEDURE IF EXISTS OttieniSpecieChePiuSiAmmala;
DELIMITER $$
CREATE PROCEDURE OttieniSpecieChePiuSiAmmala()

```

```
BEGIN
  CREATE OR REPLACE VIEW `SpecieChePiuSiAmmalano` AS
    SELECT `Nome` AS `SpecieCheSiAmmalanoDiPiu`
    FROM `specie`
    WHERE `NumeroEsordi` = (SELECT MAX(`NumeroEsordi`)
                            FROM `specie`);

  SELECT *
  FROM `SpecieChePiuSiAmmalano`;
END $$
DELIMITER ;
```