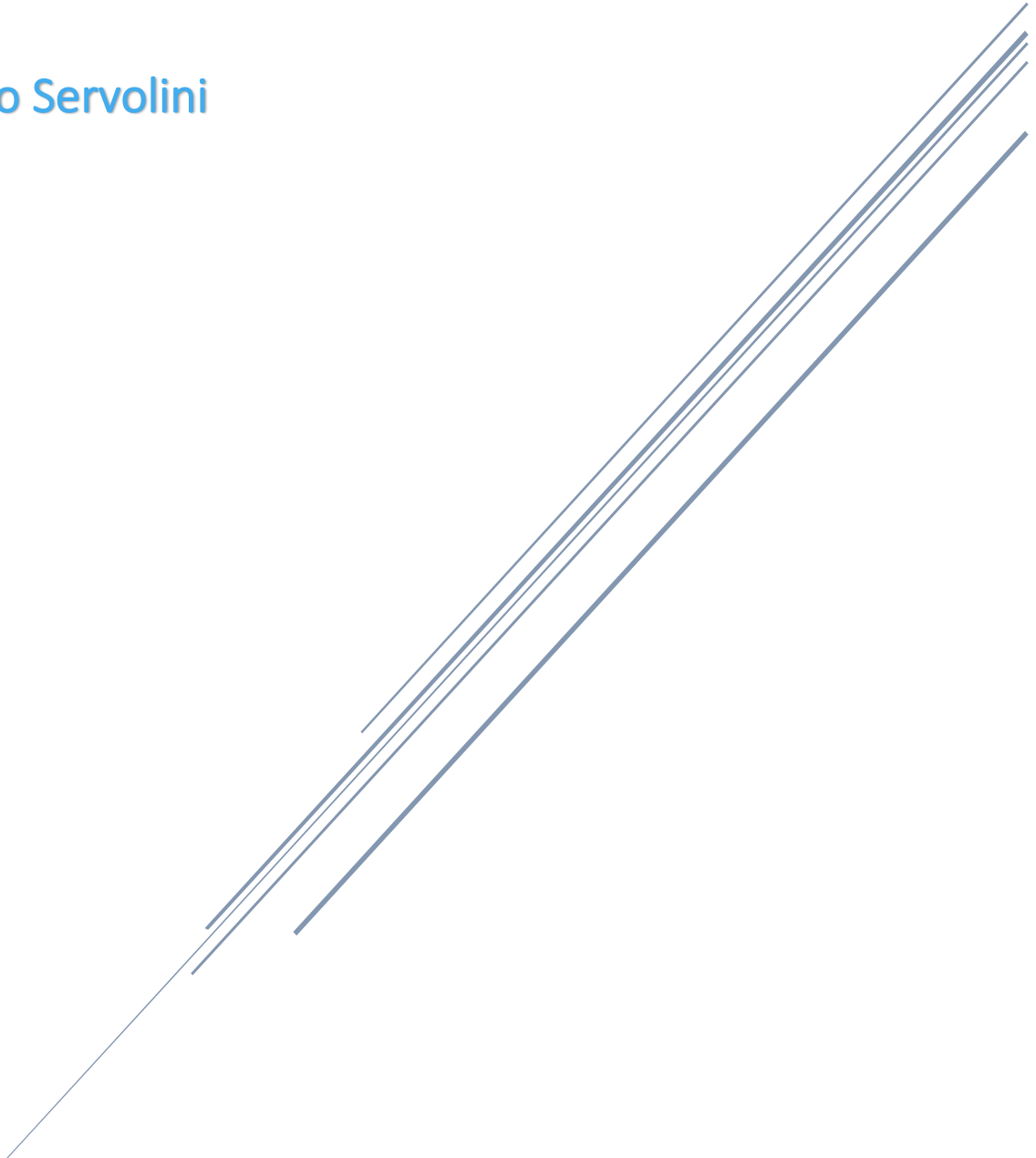




RELAZIONE DI PROGETTO PER IL CORSO DI BASI DI DATI

Anno accademico 2018/2019

Lorenzo Servolini



Sommario

1. Analisi del database e vista d'insieme.....	3
2. Glossario	4
2.1) Area allevamento	4
2.2) Area produzione	5
2.3) Area healthcare	7
2.4) Area soggiorno.....	7
2.5) Area store	8
3. Progettazione concettuale	9
3.1) Generalizzazioni.....	9
3.2) Attributi composti	10
3.3) Attributi multivalore.....	11
4. Ristrutturazione diagramma E-R	12
4.1) Traduzione delle generalizzazioni	12
4.2) Eliminazione degli attributi composti.....	14
4.3) Eliminazione degli attributi multivalore	15
4.4) Ulteriori modifiche	15
4.5) Dizionario dei dati.....	17
4.5.1) Area allevamento.....	17
4.5.2) Area produzione	19
4.5.3) Area healthcare.....	21
4.5.4) Area soggiorno	22
4.5.5) Area store.....	23
5. Individuazione delle operazioni e analisi delle prestazioni.....	25
5.1) Tavola dei volumi.....	25
5.1.1) Area Allevamento	25
5.1.2) Area Produzione	28
5.1.3) Area Healthcare	30
5.1.4) Area Soggiorno.....	31
5.1.5) Area Store	32
5.2) Individuazione delle operazioni significative e tavola degli accessi.....	33
5.2.1) Camere da preparare	33
5.2.2) Stagionatura formaggi	34
5.2.3) Report vendite	36

5.2.4) Rilevamento posizione	37
5.2.5) Saldo soggiorno.....	38
5.2.6) Aggiunta prodotto ad un ordine pendente	40
5.2.7) Tentativo di riproduzione	42
5.2.8) Animali sotto terapie	43
5.2.9) Inserimento visita	45
5.3) Introduzione di ridondanze	46
5.3.1) Camere da preparare.....	46
5.3.2) Stagionatura formaggi	47
5.3.3) Animali sotto terapie	49
5.4) Operazioni di impatto sulle ridondanze	51
5.4.1) Assegnazione camere	51
5.4.2) Lettura camera.....	51
5.4.3) Inserimento prodotto caseario	52
5.4.4) Aggiornamento scadenza	53
5.4.5) Inserimento terapia	53
5.4.6) Storico terapie.....	53
6. Traduzione verso il modello relazionale	55
6.1) Modello relazionale	55
6.2) Vincoli di integrità referenziale	57
6.3) Vincoli di integrità generici.....	62
7. Analisi delle dipendenze funzionali e normalizzazione	63
7.1) Dipendenze funzionali	63
8. Analytics	71
8.1) Comportamento degli animali.....	71
8.1.1) Spostamento medio.....	71
8.1.2) Distanza dagli altri animali.....	72
8.2) Materialized view	73
9. Implementazione su DBMS	73

1. Analisi del database e vista d'insieme

L'obiettivo del progetto è la creazione di una base di dati per memorizzare e gestire dati a supporto delle funzionalità di **Farmhouse 4.0**, una grande azienda che gestisce agriturismi di ultima generazione, altamente tecnologici e informatizzati, conformi al nuovo modello dell'industria intelligente **Industry 4.0**. Grazie a questi moderni sistemi di gestione, tutte le attività dell'agriturismo sono svolte o assistite da strutture robotizzate e automatizzate. Queste tecnologie, oltre a migliorare la qualità della vita dei dipendenti, possono aiutare a garantire il benessere degli animali, riducendo notevolmente la mole di lavoro e la fatica per i dipendenti. Di conseguenza, si possono realizzare ingenti risparmi economici e anche ridurre al minimo gli sprechi. I clienti inoltre possono soggiornare nelle camere che dispone ogni agriturismo ed è anche possibile acquistare i prodotti tramite un efficiente store web.

Per garantire una maggiore chiarezza per il glossario (*par. 2*) e per l'organizzazione strutturale dell'azienda, si è scelto di mantenere, seguendo la documentazione originale, il progetto diviso in 5 aree tematiche.

- *Area allevamento*
- *Area healthcare*
- *Area produzione*
- *Area soggiorno*
- *Area store*

Il **glossario** illustrato nel paragrafo seguente è stato realizzato dopo il diagramma Entità - Relazioni per avere una visione complessiva migliore. In esso vengono indicate le **entità** e i **termini principali**, una breve **descrizione**, dei **sinonimi** con cui è possibile riferirsi a tali entità ed eventuali **collegamenti** con altre.

2. Glossario

2.1) Area allevamento

Termine	Descrizione	Sinonimi	Collegamenti
Agriturismo	Struttura dove avviene l'allevamento degli animali e tutte le attività annesse (come il pascolo o la mungitura), l'affitto delle camere e la vendita dei prodotti artigianali	Azienda	Stalla, Zona pascolo, Camera, Cantina, Magazzino
Stalla	Fabbricato che compone l'agriturismo suddiviso in locali che ospitano gli animali		Agriturismo, Locale
Locale	Ospita gli animali di una stessa specie in base alle dimensioni e all'allestimento di cui è dotato		Stalla, Igiene, Mangiatoia, Abbeveratoio, Condizionatore, Dispositivo di illuminazione, Animale
Igiene	Livello di pulizia dei locali misurato dai sensori	Pulizia, Livello di sporcizia	Locale
Mangiatoia	Recipiente per la somministrazione automatica del foraggio		Locale, Foraggio
Abbeveratoio	Recipiente contenente costantemente acqua eventualmente arricchita con sali minerali e vitamine		Locale, Sostanza chimica
Condizionatore	Impianto che fa parte dell'allestimento dei locali		Locale
Dispositivo di illuminazione	Dispositivo che fa parte dell'allestimento dei locali		Locale
Foraggio	Alimento di ogni pasto con specifici valori nutrizionali	Alimentazione, Cibo, Fieno, Insilato	Mangiatoia
Sostanza chimica	Sostanza disciolta negli abbeveratoi dei locali		Abbeveratoio
Fornitore	Si occupa di vendere gli animali ai vari agriturismi		Animale
Zona pascolo	Zona delimitata da recinzioni dove è possibile svolgere le attività di pascolo degli animali	Zona agriturismo, Area di pascolo	Recinzione, Animale, Agriturismo

Recinzione	Delimita le zone di pascolo dell'agriturismo		Zona pascolo
Animale	Animale allevato nei locali dell'agriturismo. Può essere stato acquistato o nato nell'azienda	Capi, Gestante	Specie, Locale, Riproduzione, Gestazione, Fornitore, Posizione, Zona pascolo, Mungitrice, Latte, Visita
Specie	"Categoria" degli animali		Animale
Posizione	Latitudine e longitudine degli animali rilevate dai sensori durante il pascolo	Posizione GPS	Zona pascolo, Animale
Riproduzione	Tentativo di riproduzione supervisionato da un veterinario		Veterinario, Animale, Gestazione
Gestazione	Periodo durante il quale la gestante è sottoposta a controlli programmati. La gestazione potrebbe essere interrotta a seguito di complicanze	Scheda di gestazione	Animale, Visita, Controllo, Veterinario, Riproduzione
Controllo	Controlli eseguiti durante la gestazione	Interventi di controllo	Gestazione, Veterinario, Terapia, Esame
Esame	Esame diagnostico di accertamento richiesto da un veterinario		Controllo

2.2) Area produzione

Termine	Descrizione	Sinonimi	Collegamenti
Mungitrice	Munge gli animali nelle zone dove amano stare per garantirne il benessere		Latte, Animale
Latte	Latte ricavato dalla mungitura degli animali che verrà inviato ai silos		Mungitrice, Animale, Caratteristica, Silos
Caratteristica	Informazioni chimico-fisiche del latte munto dagli animali	Informazioni chimico-fisiche, Sostanza, Composizione chimico-fisica	Latte
Silos	Conserva il latte con composizioni chimico-fisiche simili prima di essere utilizzato dai laboratori di produzione		Agriturismo, Latte, Laboratorio di produzione

Laboratorio di produzione	Produce i prodotti caseari in lotti che verranno venduti nello store	Laboratorio	Agriturismo, Silos, Lotto
Prodotto caseario	Unità di prodotto che compongono un lotto	Prodotto, Unità di prodotti caseari, Unità di prodotto, Unità di lotto, Reso	Lotto, Formaggio, Ordine, Scaffalatura
Formaggio	Tipologie di formaggi prodotti negli agriturismi	Prodotto caseario, Tipologia di formaggio	Prodotto caseario, Ricetta, Ordine
Ricetta	Ricetta, divisa in fasi, contenente la descrizione per la produzione dei prodotti caseari		Formaggio, Fase
Fase	Fase di una ricetta accompagnata da un insieme di valori ideali di parametri di processo		Parametro di processo
Parametro di processo	Valori ideali di parametri ai quali il processo produttivo dei formaggi deve avvicinarsi il più possibile	Parametro ideale	Fase
Operaio	Dipendenti dell'agriturismo che lavorano nei laboratori di produzione	Dipendente	Lotto
Lotto	Insieme di unità di prodotti caseari prodotte utilizzando il latte con una simile composizione chimico-fisica	Lotto di produzione	Laboratorio di produzione, Parametro effettivo, Operaio, Prodotto caseario
Parametro effettivo	Valori effettivi dei parametri del processo produttivo dei prodotti caseari	Parametro	Lotto
Magazzino	Deposito in cui vengono stoccati i prodotti caseari in attesa di essere venduti		Scaffalatura, Agriturismo
Cantina	Locale adibito alla stagionatura dei formaggi		Scaffalatura, Agriturismo
Scaffalatura	Scaffale di cui sono dotati i magazzini e le cantine per permettere lo stoccaggio dei formaggi in attesa della stagionatura o della vendita		Cantina, Magazzino, Prodotto caseario

2.3) Area healthcare

Termine	Descrizione	Sinonimi	Collegamenti
Visita	Visite di controllo svolte regolarmente per monitorare lo stato di salute degli animali		Disturbo, Lesione, Patologia, Terapia, Gestazione, Veterinario, Animale
Veterinario	Supervisiona i tentativi di riproduzione, le gestazioni e la salute degli animali programmando e svolgendo visite di controllo e prescrivendo terapie	Responsabile, Personale	Terapia, Controllo, Gestazione, Riproduzione, Visita, Agriturismo
Terapia	Assunzione di farmaci o integratori per combattere e curare le patologie contratte		Controllo, Veterinario, Farmaco, Patologia, Visita
Lesione	Ferita rilevata durante una visita di controllo	Indicatore soggettivo	Visita
Disturbo	Disturbo comportamentale rilevato durante una visita di controllo	Disturbo comportamentale, Indicatore soggettivo	Visita
Patologia	Malattia o carenza rilevata durante una visita di controllo	Carenza	Visita, Terapia
Farmaco	Farmaco somministrato durante una terapia	Integratore	Terapia

2.4) Area soggiorno

Termine	Descrizione	Sinonimi	Collegamenti
Camera	Stanza semplice o suite che i clienti dell'agriturismo possono affittare	Stanza, Suite	Agriturismo, Prenotazione, Letto
Letto	Letto singolo o matrimoniale presente nelle camere dell'agriturismo		Camera
Servizio aggiuntivo	Servizi extra disponibili solo ai clienti che hanno prenotato una suite	Servizio	Prenotazione
Prenotazione	Prenotazione di una camera effettuata sul sito web dell'agriturismo	Soggiorno	Camera, Servizio aggiuntivo, Pagamento, Escursione, Cliente

Pagamento	Pagamento in contanti o con carta di credito/debito della camera e degli eventuali servizi aggiuntivi richiesti		Prenotazione
Cliente	Persona che ha prenotato una camera o effettuato acquisti sul sito web dell'agriturismo	Utente, Utente non registrato, Account	Prenotazione, Ordine
Escursione	Visita turistica della tenuta dell'agriturismo accompagnata da una guida		Guida, Prenotazione, Area tenuta
Guida	Guida che accompagna le escursioni		Agriturismo, Escursione
Area tenuta	Aree dell'agriturismo che possono essere visitate durante le escursioni		Agriturismo, Escursione

2.5) Area store

Termine	Descrizione	Sinonimi	Collegamenti
Ordine	Acquisto di prodotti caseari mediante un account sul sito web dell'agriturismo	Acquisto	Cliente, Recensione, Spedizione, Formaggio, Prodotto caseario
Spedizione	Merce acquistata spedita all'indirizzo del cliente		Centro smistamento, Ordine
Centro smistamento	Centro di smistamento della merce	Hub	Spedizione
Recensione	Valutazione facoltativa dei prodotti acquistati da un cliente		Ordine

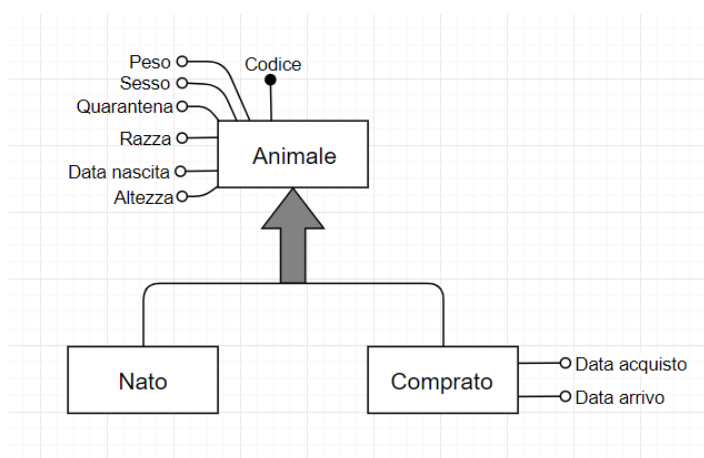
3. Progettazione concettuale

In questo capitolo viene descritta la progettazione concettuale della base di dati e il diagramma E-R prima della ristrutturazione (*Diagramma E-R ver. 1*) nelle sue varie componenti come attributi composti, generalizzazioni, attributi multivalore ecc.

Per la progettazione del diagramma Entità-Relazioni si è scelto di adottare una **strategia mista**: la strategia *bottom-up* è stata applicata nella divisione del database in aree tematiche, poi sviluppate per raffinamenti successivi (*top-down*). Questa scelta deriva sia dall'esigenza di suddividere il lavoro, in quanto si parla di uno schema complesso che comprende molte entità e relazioni, sia dalla necessità di avere una visione generale per poter raffinare e definire lo schema nei dettagli (vantaggio della strategia *top-down*).

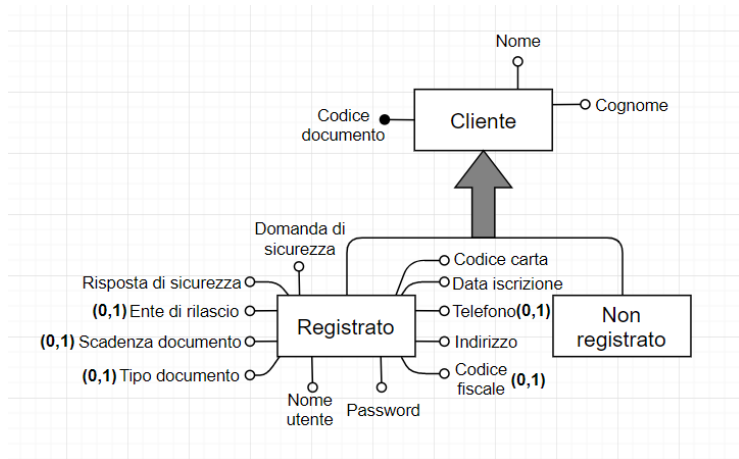
3.1) Generalizzazioni

Due concetti presenti nello schema sono stati rappresentati come generalizzazioni; questa scelta di progettazione è stata presa in seguito a considerazioni relative alla natura delle entità stesse e dei loro attributi molto simili.

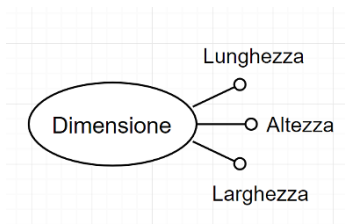


Gli animali sono suddivisi in Nati e Comprati. Per entrambi abbiamo gli stessi dati con la differenza che quelli comprati avranno anche una data di acquisto ed una di arrivo. La generalizzazione è **totale** poiché ogni animale sarà obbligatoriamente o acquistato o nato all'interno dell'agriturismo ed **esclusiva** in quanto se un animale è stato comprato da un fornitore non può anche essere nato nell'agriturismo e viceversa.

I clienti sono suddivisi in Registrati e Non registrati, principalmente perché, nonostante entrambi abbiano in comune i dati relativi al riconoscimento della persona, per i primi vengono indicati anche i dati relativi all'account e al documento d'identità. La generalizzazione è **totale** poiché ogni cliente che accede al sito web sarà obbligatoriamente registrato o non registrato ed **esclusiva** in quanto se un cliente è registrato non potrà essere anche non registrato e viceversa.

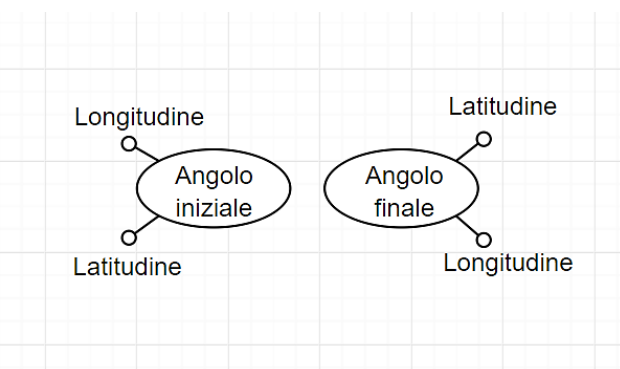
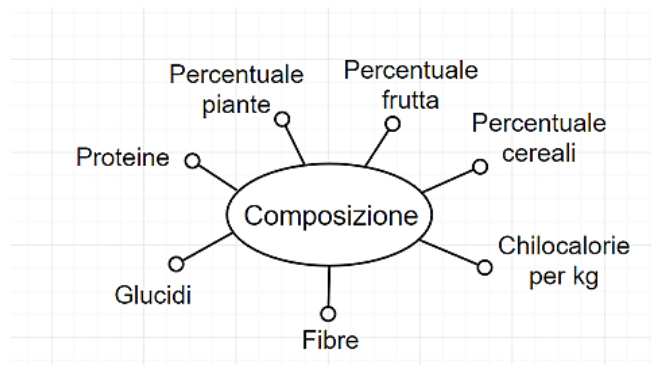


3.2) Attributi composti



Per indicare le dimensioni del Locale è stato creato un attributo composto in quanto per fornire indicazioni esatte è necessario indicare più parametri (Lunghezza, Larghezza, Altezza).

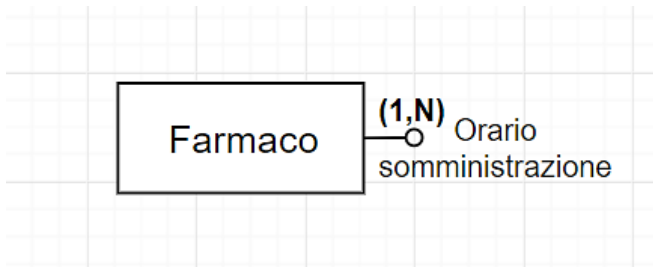
È stato creato l'attributo composto "Composizione" per indicare le percentuali delle diverse componenti e i valori nutrizionali di un tipo di Foraggio, utilizzato all'interno dell'agriturismo.



Sono stati creati due attributi composti (Angolo iniziale e Angolo finale) per fornire le indicazioni sulla posizione dei due angoli delle recinzioni (quadrate) che compongono le aree di pascolo.

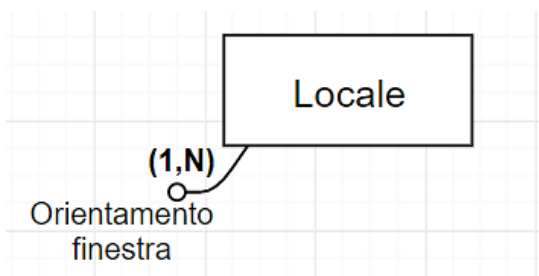
3.3) Attributi multivalore

- Entità Farmaco



Per indicare i vari orari giornalieri in cui un farmaco deve essere somministrato è stato creato un attributo multivalore.

- Entità Locale



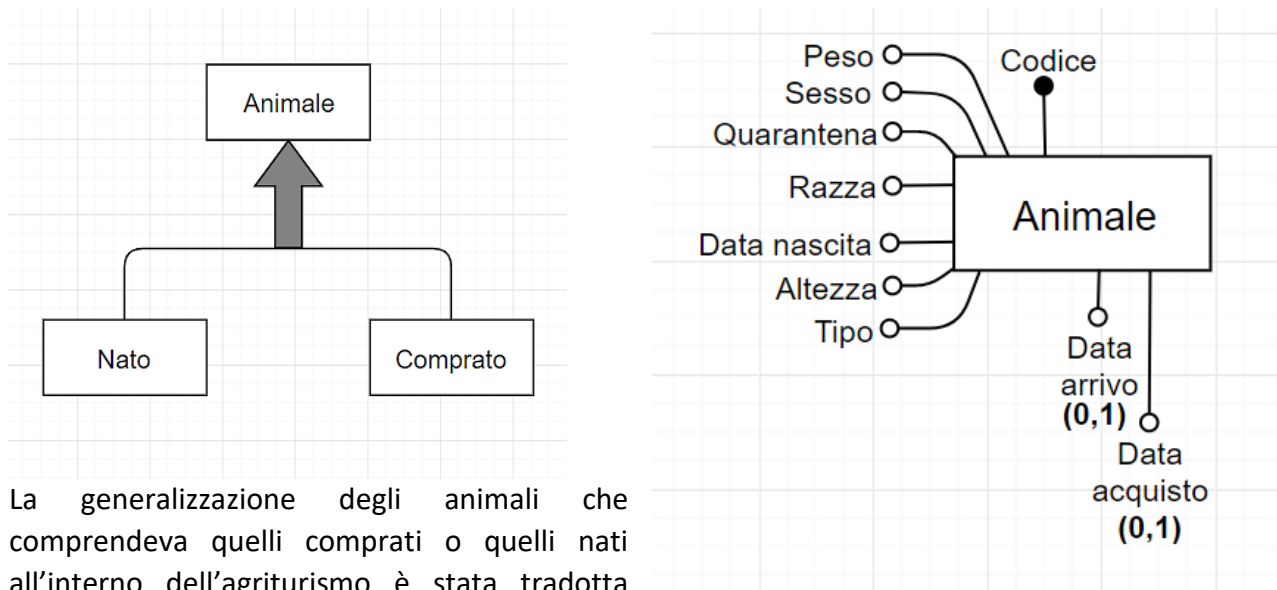
Per poter specificare i possibili orientamenti cardinali delle finestre presenti all'interno di un locale è stato creato un attributo multivalore.

4. Ristrutturazione diagramma E-R

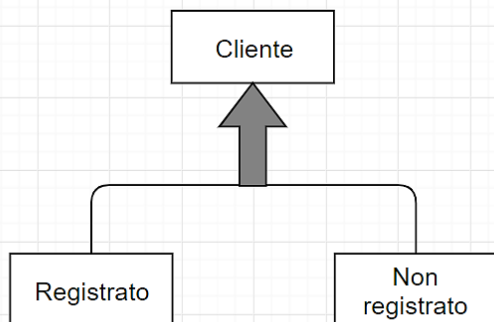
In questo capitolo viene trattata la prima parte della progettazione logica della base di dati, ossia la ristrutturazione dello schema E-R. Verranno sostituite le due generalizzazioni e gli attributi composti o multivalore in quanto non sono direttamente traducibili nel modello logico relazionale e, infine, verranno apportate delle modifiche al diagramma ristrutturato.

Al termine di tutte le operazioni sarà redatto il dizionario dei dati (*cap. 4.5*).

4.1) Traduzione delle generalizzazioni

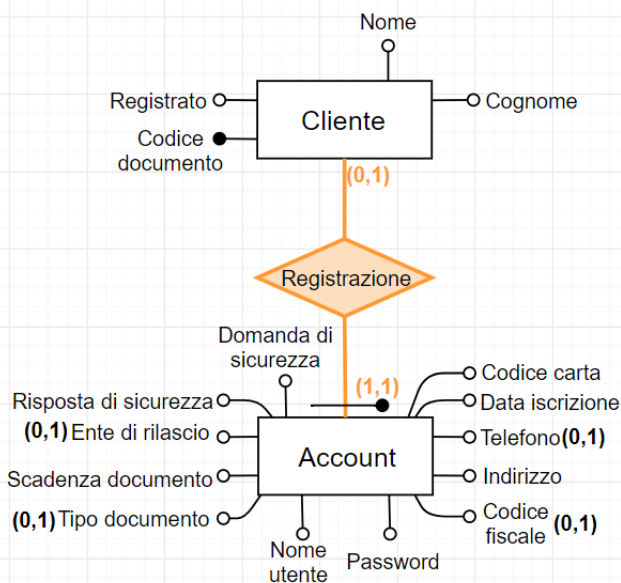


La generalizzazione degli animali che comprendeva quelli comprati o quelli nati all'interno dell'agriturismo è stata tradotta eliminando le entità figlie e lasciando solo l'entità padre, assegnando a quest'ultima tutti gli attributi delle altre due; nell'entità padre è stato inoltre aggiunto l'attributo **Tipo**, utile a capire a quale *categoria* (nato o comprato) l'animale appartenga. La scelta di mantenere una sola entità è dovuta al fatto che le relazioni interessano praticamente solo il padre, sarebbe quindi stato sconveniente mantenere due entità *specifiche* (Nato e Comprato) invece di una *generica* (Animale).



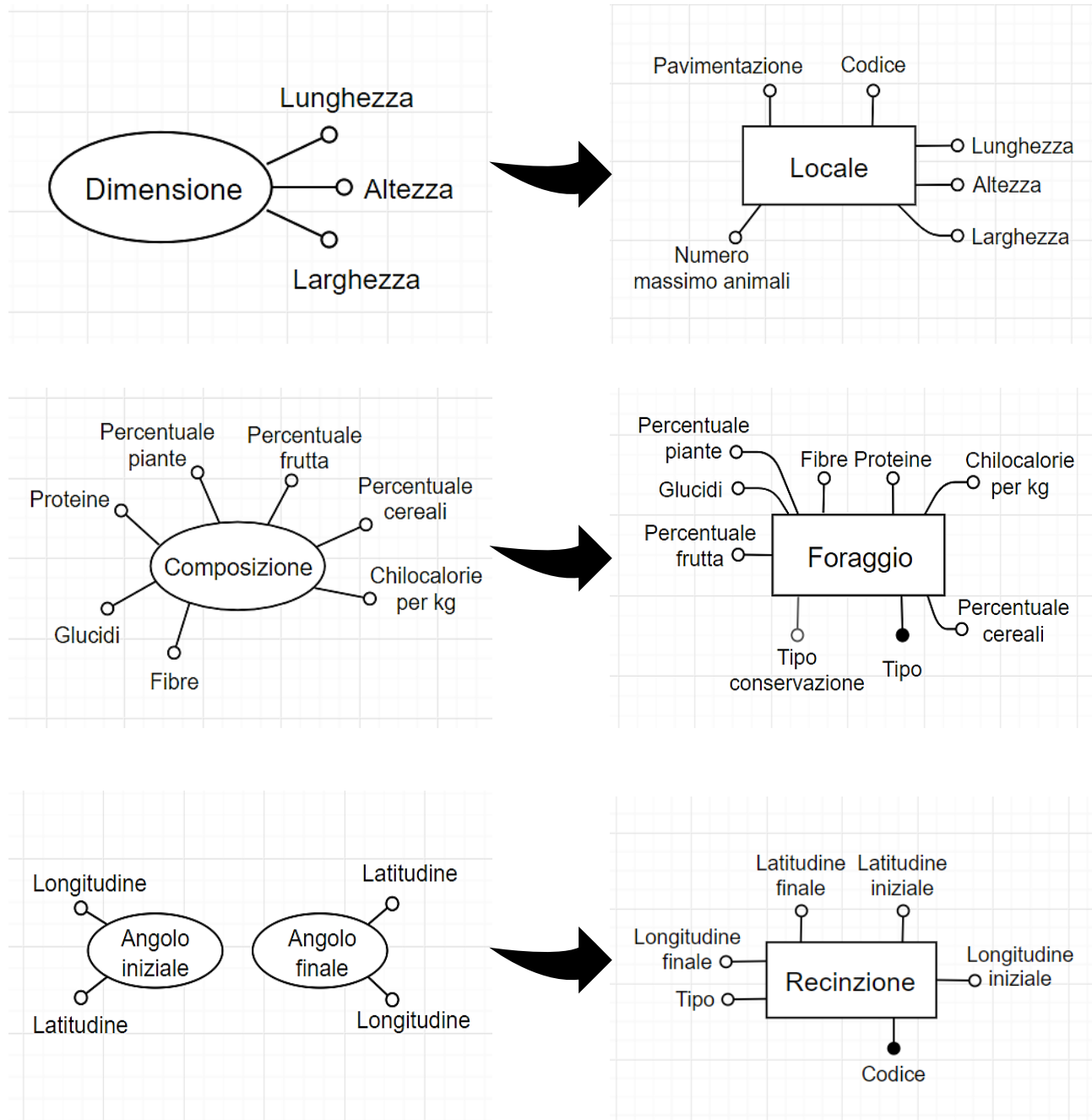
La generalizzazione dei clienti che comprendeva quelli registrati e quelli non registrati è stata tradotta con una soluzione ibrida: è stata eliminata l'entità figlia Non registrato ed è stata inserita una relazione tra

Cliente e Registrato (il cui nome è stato cambiato in Account per maggiore chiarezza); nell'entità padre è stato inoltre aggiunto l'attributo booleano **Registrato**, utilizzato per mantenere la distinzione che la generalizzazione rappresentava. La scelta di una traduzione mista è derivata dal fatto che l'entità figlia Non registrato non aveva relazioni né attributi *aggiuntivi* e l'entità figlia Registrato aveva delle operazioni che la avrebbero interessata separatamente dall'entità padre.

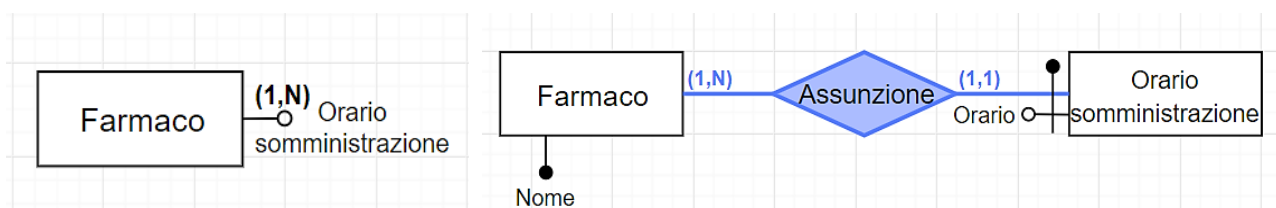


4.2) Eliminazione degli attributi composti

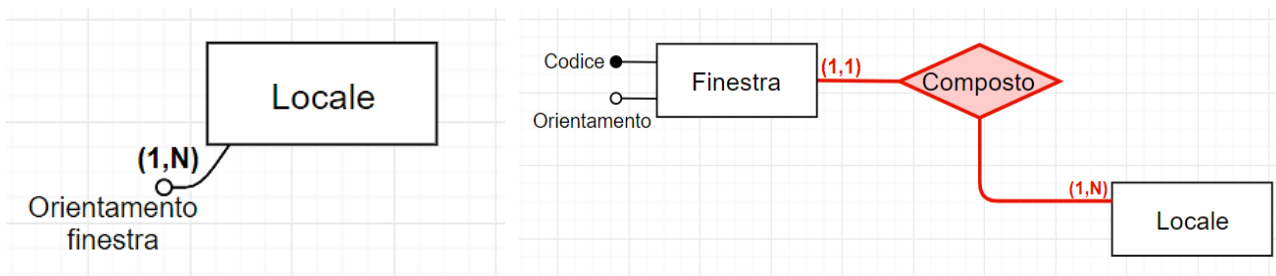
Gli attributi composti sono tutti stati sostituiti con i singoli attributi componente sull'entità a cui erano *collegati*.



4.3) Eliminazione degli attributi multivalore



L'attributo multivalore Orario somministrazione posseduto dalle entità Farmaco è stato sostituito con un'entità omonima. Tale entità è stata poi messa in relazione con Farmaco, in modo da poter risalire al farmaco a cui gli orari si riferiscono.



L'attributo multivalore Orientamento finestra posseduto dalle entità Locale è stato sostituito con l'entità Finestra. Tale entità è stata poi messa in relazione con Locale, in modo da poter risalire ai locali dove quelle finestre sono posizionate.

4.4) Ulteriori modifiche

Al termine delle operazioni di ristrutturazione sono state apportate piccole modifiche agli attributi, alle entità e alle relazioni, ottenendo quindi il diagramma E-R nella sua forma finale.

- Spostamento dell'attributo **Gravità** dall'entità **Patologia** alla relazione **Esordio** in quanto ogni patologia non colpirà mai un animale sempre con la stessa gravità.
- Rinominati gli attributi **Data e ora**/**Ora lettura**/**Orario riproduzione** delle entità **Pagamento**, **Ordine**, **Visita**, **Stato abbeveratoio**, **Stato mangiatoia**, **Intervento pulizia**, **Posizione**, **Stato cantina**, **Stato aria**, **Riproduzione** con il nome **Istante** per maggiore chiarezza.
- Rinominato l'attributo **Dose giornaliera** dell'entità **Farmaco** con il nome **Posologia** per una maggiore precisione.
- È stato aggiunto l'identificatore esterno sull'entità **Camera** in quanto ogni numero identificatore delle camere è unico solo all'interno di uno specifico agriturismo.
- Aggiunta dell'attributo **Portata** e **Capacità formaggi** alla relazione **Scaffalatura** per specificare la portata in kg e il numero massimo di formaggi che una scaffalatura può sostenere.
- Creazione della relazione **Proprietà** tra **Agriturismo** e **Laboratorio di produzione** per specificare a quale agriturismo appartenga un laboratorio.

- Creazione della relazione **Ubicazione** tra **Agriturismo** e **Silos** per specificare a quale agriturismo appartenga un silos.
- Creazione della relazione **Situata** tra **Agriturismo** e **AreaTenuta** per specificare in quale agriturismo si trovi quell'area.
- Creazione della relazione **Afferenza** tra **Agriturismo** e **Veterinario** per specificare l'agriturismo dove lavora un veterinario.
- Creazione della relazione **Afferenza guida** tra **Agriturismo** e **Guida** per specificare l'agriturismo dove lavora una guida.
- Creazione della relazione **Alloggia** tra **Locale** e **Specie** per specificare la specie che ospita.
- Spostamento dell'attributo **Numero massimo animali** dall'entità **Locale** alla relazione **Alloggia** in quanto ogni il numero massimo di animali dipende dalle dimensioni del locale ma anche dalla specie.
- Creazione dell'entità **Indicatore benessere** con la relativa relazione **Monitoraggio** che la collega a **Visita** per poter rimuovere tutti gli attributi opzionali da **Visita**.
- Spostamento degli attributi **Data acquisto** e **Data arrivo** dall'entità **Animale** alla relazione **Acquisto** in quanto sono informazioni relative alla compravendita.
- Spostamento dell'attributo **Codice carta** dall'entità **Pagamento** all'entità **Cliente** in quanto la carta è richiesta sia per i clienti registrati che per i non registrati ma non nei pagamenti (in quanto potrebbero essere effettuati in contanti). Come conseguenza l'attributo **Codice carta** è stato eliminato dall'entità **Account** in quanto ridondante.
- Aggiunta dell'attributo **Importo** sull'entità **Pagamento** per tenere traccia delle somme pagate.
- Aggiunta dell'attributo **Prezzo al kg** sull'entità **Formaggio**.
- Aggiunta dell'attributo **Sconto** sull'entità **Prodotto caseario** per poter applicare degli sconti.
- Eliminato l'attributo **Codice** dall'entità **Visita** e aggiunto l'identificatore esterno sull'attributo **Istante** e l'associazione **Soggetto** per specificare che un animale possa essere visitato più volte in una giornata ma non nello stesso momento.
- Crea l'entità **Documento** con la relativa relazione **Possiede** che la collega ad **Account** per rimuovere gli attributi opzionali da **Account**.
- Crea l'entità **Scheda valutazione** con la relativa relazione **Reso** che la collega a **Prodotto caseario** per poter registrare i parametri di qualità analizzati dall'azienda del prodotto reso.
- Eliminato l'attributo **Reso** sull'associazione **Fornitura**.
- Crea l'entità **Complicanza** con la relativa relazione **Interruzione** che la collega a **Gestazione** per rimuovere gli attributi opzionali da **Gestazione**.

4.5) Dizionario dei dati

Di seguito è illustrata la tabella contenente tutte le entità e relazioni del database, per le prime sono stati specificati gli attributi e gli identificatori, mentre per le seconde una breve descrizione, le entità che mettono in relazione con la relativa cardinalità ed infine eventuali attributi.

4.5.1) Area allevamento

4.5.1.1) Entità

Entità	Attributi	Identificatore
Abbeveratoio	Codice, Capienza	Codice
Agriturismo	Codice, Indirizzo	Codice
Animale	Codice, Peso, Sesso, Quarantena, Razza, Data nascita, Altezza, Tipo	Codice
Complicanza	Descrizione, Data	Gestazione (esterno)
Condizionatore	Codice, Classe	Codice
Controllo	Codice, Stato, Data programmata, Data effettiva, Esito	Codice
Dispositivo illuminazione	Codice, Lumen, Grado protezione	Codice
Esame	Codice, Nome, Data, Macchinario, Descrizione	Codice
Finestra	Codice, Orientamento	Codice
Foraggio	Tipo, Percentuale piante, Percentuale cereali, Percentuale frutta, Chilocalorie per kg, Proteine, Fibre, Glucidi, Tipo conservazione	Tipo
Fornitore	Partita IVA, Ragione sociale, Nome, Indirizzo	Partita IVA
Gestazione	Codice	Codice
Intervento pulizia	Istante, Stato pulizia, Livello sporcizia, Livello azoto, Livello metano	Istante, Locale (esterno)
Locale	Codice, Pavimentazione, Lunghezza, Larghezza, Altezza	Codice
Mangiatoia	Codice, Larghezza, Lunghezza	Codice
Posizione	Istante, Latitudine, Longitudine	Istante, Animale (esterno)
Recinzione	Codice, Tipo, Longitudine iniziale, Longitudine finale, Latitudine iniziale, Latitudine finale	Codice
Riproduzione	Istante, Stato	Istante, Padre (esterno), Madre (esterno)
Sostanza chimica	Nome, Vitamine, Sali minerali	Nome
Specie	Nome, Famiglia	Nome
Stalla	Codice, Zona	Codice
Stato abbeveratoio	Istante, Quantità acqua	Istante, Abbeveratoio (esterno)
Stato aria	Istante, Temperatura, Umidità	Istante, Locale (esterno)

Stato mangiatoia	Istante, Quantità cibo	Istante, Mangiatoia (esterno)
Veterinario	Codice, Nome, Cognome, Telefono	Codice
Zona pascolo	Nome	Nome, Agriturismo (esterno)

4.5.1.2) Relazioni

Relazioni	Descrizione	Entità Coinvolte	Attributi
Acquisto	Associa un animale al fornitore da cui è stato acquistato	Fornitore (0,N) Animale (0,1)	Data arrivo, Data acquisto
Afferenza	Associa un veterinario all'agriturismo in cui lavora	Agriturismo (1,N) Veterinario (1,1)	
Allestimento abbeveratoio	Associa un locale ai suoi abbeveratoi	Locale (1,N) Abbeveratoio (1,1)	
Allestimento condizionatore	Associa un locale ai suoi condizionatori	Locale (1,N) Condizionatore (1,1)	
Allestimento illuminazione	Associa un locale ai suoi impianti di illuminazione	Locale (1,N) Dispositivo illuminazione (1,1)	
Allestimento mangiatoia	Associa un locale alle sue mangiatoie	Locale (1,N) Mangiatoia (1,1)	
Alloggia	Associa un locale alla specie che ospita	Locale (1,1) Specie (0,N)	Numero massimo animali
Appartiene	Associa un animale alla propria specie	Specie (0,N) Animale (1,1)	
Comporta	Associa un controllo alla terapia prescritta in seguito al controllo	Controllo (0,N) Terapia (0,1)	
Composto	Associa una finestra al locale dove è situata	Locale (1,N) Finestra (1,1)	
Contiene	Associa lo stato di una mangiatoia al tipo di foraggio rilevato	Stato mangiatoia (0,1) Foraggio (0,N)	
Controllo aria	Associa le informazioni sull'aria al locale dove è stata effettuata la misurazione	Locale (1,N) Stato aria (1,1)	
Costituito	Associa una stalla all'agriturismo a cui appartiene	Agriturismo (1,N) Stalla (1,1)	
Delimitazione	Associa una zona di pascolo alla sua recinzione	Zona pascolo (1,N) Recinzione (1,1)	
Dispone	Associa una zona di pascolo al relativo agriturismo	Agriturismo (1,N) Zona pascolo (1,1)	
Divisa	Associa un locale alla sua stalla	Stalla (1,N) Locale (1,1)	

Interruzione	Associa una gestazione alle sue complicanze che hanno portato ad interromperla	Gestazione (0,1) Complicanza (1,1)	
Nascita	Associa una gestazione all'animale partorito	Gestazione (0,N) Animale (0,1)	
Necessario	Associa un controllo all'esame richiesto in seguito al controllo	Esame (1,1) Controllo (0,N)	
Ospita	Associa un animale al locale che lo ospita	Locale (0,N) Animale (1,1)	
Pascolo	Associa un animale alla zona dove sta pascolando e alla sua posizione attuale	Animale (0,N) Posizione (1,1)	
Proprietà acqua	Associa lo stato di un abbeveratoio alle sostanze chimiche rilevate	Stato abbeveratoio (0,N) Sostanza chimica (0,N)	
Responsabile	Associa una gestazione al veterinario responsabile	Gestazione (1,1) Veterinario (0,N)	
Richiesta	Associa un intervento di pulizia al locale che deve essere pulito	Locale (1,N) Intervento pulizia (1,1)	
Sensore abbeveratoio	Associa lo stato di un abbeveratoio all'abbeveratoio dove è stata effettuata la misurazione	Abbeveratoio (1,N) Stato abbeveratoio (1,1)	
Sensore mangiatoia	Associa lo stato di una mangiatoia alla mangiatoia dove è stata effettuata la misurazione	Mangiatoia (1,N) Stato mangiatoia (1,1)	
Sottoposta	Associa una gestazione ai controlli a cui è sottoposta	Gestazione (1,N) Controllo (1,1)	
Successo	Associa una riproduzione che ha avuto successo alla relativa gestazione	Gestazione (1,1) Riproduzione (0,1)	
Supervisiona	Associa un tentativo di riproduzione al veterinario supervisore	Riproduzione (1,1) Veterinario (0,N)	
Svolge	Associa un controllo al veterinario che lo ha svolto	Controllo (1,1) Veterinario (0,N)	
Tentativo	Associa un tentativo di riproduzione ai due animali che si sono accoppiati	Animale (0,N) Riproduzione (1,1)	

4.5.2) Area produzione

4.5.2.1) Entità

Entità	Attributi	Identificatore
Cantina	Codice, Classe energetica	Codice
Caratteristica	Sostanza, Quantità mg/ml	Sostanza, Latte (esterno)

Fase	Fase, Descrizione	Fase, Formaggio (esterno)
Formaggio	Tipo, Zona d'origine, Tipo pasta, Deperibilità, Prezzo al kg	Tipo
Laboratorio produzione	Codice, Superficie, Numero vasche	Codice
Latte	Codice, Volume	Codice
Lotto	Codice, Data, Scadenza, Durata produzione	Codice
Magazzino	Codice, Superficie, Classe energetica	Codice
Mungitrice	Codice, Latitudine, Longitudine	Codice
Operaio	Codice, Nome, Cognome, Stipendio	Codice
Parametro di processo	Nome parametro, Valore	Nome parametro, Fase (esterno), Formaggio (esterno)
Parametro effettivo	Fase, Nome parametro, Valore	Fase, Nome parametro, Lotto (esterno)
Prodotto caseario	Codice, Peso, Sconto	Codice
Scaffalatura	Codice, Marca, Portata, Capacità formaggi	Codice
Silos	Codice, Livello di riempimento, Capacità ettolitri	Codice
Stato cantina	Istante, Temperatura, Umidità, Ventilazione	Istante, Cantina (esterno)

4.5.2.2) Relazioni

Relazioni	Descrizione	Entità Coinvolte	Attributi
Collocazione	Associa un prodotto caseario alla scaffalatura su cui è collocato	Prodotto caseario (1,1) Scaffalatura (0,N)	
Contenuto	Associa un silos alle informazioni sul latte munto che contiene	Latte (1,1) Silos (0,N)	
Dotato	Associa una cantina al suo agriturismo	Cantina (1,1) Agriturismo (1,N)	
Formato	Associa un lotto le unità di prodotti caseari che lo compongono	Lotto (1,N) Prodotto caseario (1,1)	
Lavora	Associa ad ogni lotto gli operai che ci hanno lavorato	Lotto (1,N) Operaio (0,N)	
Mungitura	Associa un animale alla mungitrice utilizzata per la mungitura e alle informazioni sul latte prodotto	Animale (0,N) Mungitrice (0,N) Latte (1,1)	Istante
Processo produttivo	Associa ad ogni lotto i parametri effettivi di processo	Lotto (1,N) Parametro effettivo (1,1)	

Produzione	Associa un lotto al silos da cui è stato estratto il latte utilizzato e il laboratorio che lo ha prodotto	Silos (0,N) Laboratorio produzione (0,N) Lotto (1,1)	
Proprietà	Associa un laboratorio di produzione all'agriturismo che ne detiene la proprietà	Agriturismo (1,N) Laboratorio produzione (1,1)	
Proprietà latte	Associa le informazioni del latte prodotto da una mungitura alle sue caratteristiche	Latte (1,N) Caratteristica (1,1)	
Provvisto	Associa un magazzino al suo agriturismo	Magazzino (1,1) Agriturismo (1,N)	
Ricetta	Associa ad ogni tipo di formaggio la ricetta per produrlo	Formaggio (1,N) Fase (1,1)	
Sensore cantina	Associa le informazioni sull'aria alla cantina dove è stata effettuata la misurazione	Cantina (1,N) Stato cantina (1,1)	
Stagionatura	Associa una scaffalatura alla cantina dove è posizionata	Scaffalatura (0,1) Cantina (0,N)	
Stoccaggio	Associa una scaffalatura al magazzino dove è posizionata	Scaffalatura (0,1) Magazzino (0,N)	
Tipologia	Associa un prodotto caseario al tipo di formaggio	Prodotto caseario (1,1) Formaggio (0,N)	
Ubicazione	Associa ad ogni Silos l'agriturismo in cui si trova	Agriturismo (1,N) Silos (1,1)	
Valore ideale	Associa ad ogni fase delle ricette i valori ideali dei parametri di processo	Fase (1,N) Parametro di processo (1,1)	

4.5.3) Area healthcare

4.5.3.1) Entità

Entità	Attributi	Identificatore
Disturbo	Nome, Entità	Nome, Visita (esterno)
Farmaco	Nome, Posologia, Principio attivo	Nome
Indicatore benessere	Nome	Nome
Lesione	Tipologia, Parte del corpo, Entità	Tipologia, Parte del corpo, Visita (esterno)
Orario somministrazione	Orario	Orario, Farmaco (esterno)
Patologia	Nome, Parte del corpo	Nome
Terapia	Codice, Data inizio, Durata	Codice
Visita	Istante	Istante, Animale (esterno)

4.5.3.2) Relazioni

Relazioni	Descrizione	Entità Coinvolte	Attributi
Assunzione	Associa un farmaco ai suoi orari di somministrazione	Farmaco (1,N) Orario somministrazione (1,1)	
Compie	Associa una visita al veterinario che l'ha eseguita	Visita (1,1) Veterinario (0,N)	
Esordio	Associa una visita alla patologia riscontrata e alla terapia prescritta come rimedio	Visita (0,N) Patologia (0,N) Terapia (0,1)	Gravità
Monitoraggio	Associa un indicatore alla visita durante la quale è stato misurato	Visita (0,N) Indicatore benessere (0,N)	Valore
Rilevazione	Associa un disturbo alla visita durante la quale è stato rilevato	Disturbo (1,1) Visita (0,N)	
Soggetto	Associa una visita all'animale che è stato visitato	Visita (1,1) Animale (0,N)	
Somministra	Associa una terapia ai farmaci che comprende	Terapia (1,N) Farmaco (0,N)	Giorni di pausa
Valutazione	Associa una lesione alla visita durante la quale è stata rilevata	Lesione (1,1) Visita (0,N)	

4.5.4) Area soggiorno

4.5.4.1) Entità

Entità	Attributi	Identificatore
Account	Nome utente, Password, Codice fiscale, Indirizzo, Telefono, Data iscrizione, Domanda di sicurezza, Risposta di sicurezza	Cliente (esterno)
Area tenuta	Codice, Superficie	Codice
Camera	Numero, Tipo, Costo	Numero, Agriturismo (esterno)
Cliente	Codice documento, Nome, Cognome, Registrato, Codice carta	Codice documento
Escursione	Codice, Data e ora d'inizio, Costo	Codice
Guida	Codice, Nome, Cognome	Codice
Letto	Codice, Capienza, Tipo	Codice
Pagamento	Codice, Istante, Metodo di pagamento, Importo	Codice
Prenotazione	Codice, Data arrivo, Data partenza	Codice
Servizio aggiuntivo	Servizio, Costo	Servizio

Step	Numero, Tempo di sosta	Numero, Escursione (esterno)
------	------------------------	---------------------------------

4.5.4.2) Relazioni

Relazioni	Descrizione	Entità Coinvolte	Attributi
Afferenza guida	Associa una guida all'agriturismo in cui lavora	Agriturismo (1,N) Guida (1,1)	
Affitto	Associa una camera al suo agriturismo	Agriturismo (1,N) Camera (1,1)	
Assegnazione	Associa una camera ad una prenotazione	Camera (0,N) Prenotazione (1,N)	
Associata	Associa un'escursione alla sua guida	Escursione (1,1) Guida (0,N)	
Capienza	Associa una camera ai letti collocati all'interno di essa	Camera (1,N) Letto (1,1)	
Effettua	Associa una prenotazione al cliente che l'ha effettuata	Prenotazione (1,1) Cliente (0,N)	
Itinerario	Associa un'escursione ai suoi step	Escursione (1,N) Step (1,1)	
Percorre	Associa uno step di un'escursione all'area della tenuta visitata	Area tenuta (0,N) Step (1,1)	
Prenota	Associa una prenotazione all'escursione prenotata	Prenotazione (0,N) Escursione (0,N)	
Registrazione	Associa un cliente al suo account	Cliente (0,1) Account (1,1)	
Richiede	Associa una prenotazione al servizio aggiuntivo richiesto	Prenotazione (0,N) Servizio aggiuntivo (0,N)	Numero giorni
Situata	Associa un agriturismo alle aree della tenuta in cui è suddiviso	Agriturismo (1,N) Area tenuta (1,1)	
Transazione	Associa un pagamento alla prenotazione saldata	Prenotazione (0,N) Pagamento (1,1)	

4.5.5) Area store

4.5.5.1) Entità

Entità	Attributi	Identificatore
Centro smistamento	Codice, Numero dipendenti, Latitudine, Longitudine	Codice
Documento	Tipo, Ente di rilascio, Scadenza	Account (esterno)
Ordine	Codice, Istante, Stato	Codice
Recensione	Codice, Gradimento, Gusto, Qualità, Conservazione, Note aggiuntive	Codice

Scheda valutazione	Odore, Colore, Muffa, Valutazione generale	Prodotto caseario (esterno)
Spedizione	Codice, Stato, Data consegna prevista, Data consegna effettiva	Codice

4.5.5.2) Relazioni

Relazioni	Descrizione	Entità Coinvolte	Attributi
Attesa	Associa un ordine ai tipi di formaggi ordinati non presenti in magazzino	Ordine (0,N) Formaggio (0,N)	Quantità
Esegue	Associa un ordine all'account che l'ha eseguito	Ordine (1,1) Account (0,N)	
Fornitura	Associa un ordine ai prodotti caseari che verranno spediti appena l'ordine sarà completamente pronto	Ordine (0,N) Prodotto caseario (0,1)	
Invio	Associa un ordine alla sua spedizione	Ordine (0,1) Spedizione (1,1)	
Percorso	Associa una spedizione ai centri di smistamento da cui passa	Spedizione (1,N) Centro smistamento (0,N)	
Possiede	Associa un account alle informazioni registrate sul proprio documento	Account (0,1) Documento (1,1)	
Reso	Associa un prodotto restituito ai parametri di qualità analizzati dall'azienda	Prodotto caseario (0,1) Scheda valutazione (1,1)	
Riceve	Associa un ordine alla sua recensione	Recensione (1,1) Ordine (0,1)	

5. Individuazione delle operazioni e analisi delle prestazioni

In questo paragrafo viene illustrata la tavola dei volumi, le principali operazioni significative individuate con le relative tavole degli accessi e lo studio sulle tre ridondanze inserite. La tavola dei volumi mostra una stima del carico di dati per ogni entità e relazione della base di dati. Grazie a questa è stato possibile individuare delle operazioni che avessero un forte contributo nel carico applicativo del sistema. La tavola degli accessi serve invece ad illustrare proprio il carico applicativo, in base al numero di scritture e letture necessarie.

5.1) Tavola dei volumi

5.1.1) Area Allevamento

Concetto	Tipo	Volume	Motivazione
Abbeveratoio	E	$6 * 1'600 = 9'600$	6 abbeveratoi per locale (ipotesi)
Agriturismo	E	40	Ipotesi
Animale	E	$12 * 1'600 = 19'200$	12 animali per ogni locale (ipotesi)
Complicanza	E	$8'085 * 0,05 = 405$	Il 5% delle gestazioni viene interrotto (ipotesi)
Condizionatore	E	$2 * 1'600 = 3'200$	2 condizionatori per locale (ipotesi)
Controllo	E	$8'085 * 6 = 48'510$	Per ogni gestazione sono stati fatti in media 6 controlli (ipotesi)
Dispositivo illuminazione	E	$4 * 1'600 = 6'400$	4 dispositivi di illuminazione per locale (ipotesi)
Esame	E	$48'510 / 11 = 4'410$	In media ogni 11 controlli viene prescritto un esame diagnostico (ipotesi)
Finestra	E	$2 * 1'600 = 3'200$	2 finestre per ogni locale (ipotesi)
Foraggio	E	20	Ipotesi
Fornitore	E	15	Ipotesi
Gestazione	E	$10'105 * 0,8 = 8'085$	L'80% dei tentativi di riproduzione sono terminati successo
Intervento pulizia	E	$10 * 1'600 = 16'000$	Ogni 3 giorni (10 volte al mese) viene richiesta una pulizia dei locali, tenuta in memoria per un mese (ipotesi)
Locale	E	$4 * 400 = 1'600$	4 locali per stalla (ipotesi)

Mangiatoia	E	$3 * 1'600 = 4'800$	3 mangiatoie per locale (ipotesi)
Posizione	E	$60 * 8 * 21 * 19'200 = 193'536'000$	Per ogni animale, durante le 8 ore di pascolo giornaliero, viene salvata la posizione ogni minuto, memorizzata per 3 settimane (ipotesi)
Recinzione	E	600	Una recinzione per ogni zona di pascolo
Riproduzione	E	$((19'200 * 0,4 * 100) / 95) * 100 / 80 = 10'105$	Il 40% degli animali dell'agriturismo è nato grazie alle riproduzioni. L'80% dei tentativi totali di riproduzione e il 95% delle gestazioni sono andati a buon fine (ipotesi)
Sostanza chimica	E	15	Ipotesi
Specie	E	3	Possibili specie
Stalla	E	$10 * 40 = 400$	10 stalle per ogni agriturismo (ipotesi)
Stato abbeveratoio	E	$12 * 30 * 9'600 = 3'456'000$	Ogni due ore (12 volte al giorno) vengono salvate le informazioni sull'acqua presente in ogni abbeveratoio, che rimangono memorizzate per un mese (ipotesi)
Stato aria	E	$(6 * 30) * 1'600 = 288'000$	I parametri di ogni locale vengono misurati 6 volte al giorno e memorizzati per un mese (ipotesi)
Stato mangiatoia	E	$2 * 6 * 30 * 4'800 = 1'728'000$	Per 2 pasti giornalieri, ogni ora per 6 ore, viene salvata la quantità di foraggio residua nelle mangiatoie tenuta in memoria per un mese
Veterinario	E	$19'200 / 200 = 96$	1 veterinario ogni 200 animali (ipotesi)
Zona pascolo	E	$15 * 40 = 600$	15 zone di pascolo per agriturismo (ipotesi)
Acquisto	R	11'520	Il 60% degli animali dell'agriturismo è stato acquistato
Afferenza	R	96	Un'occorrenza per ogni veterinario
Allestimento abbeveratoio	R	$1'600 * 6 = 9'600$	Ciascun locale è dotato di 6 abbeveratoi

Allestimento condizionatore	R	$1'600 * 2 = 3'200$	Ciascun locale è dotato di 2 dispositivi di condizionamento
Allestimento illuminazione	R	$1'600 * 4 = 6'400$	Ciascun locale è dotato di 4 impianti di illuminazione
Allestimento mangiatoia	R	$1'600 * 3 = 4'800$	Ciascun locale è dotato di 3 mangiatoie
Alloggia	R	1'600	Un'occorrenza per ogni locale
Appartiene	R	19'200	Una specie per ogni animale
Comporta	R	$48'510 / 14 = 3'465$	In media ogni 14 controlli è prescritta una terapia (ipotesi)
Composto	R	$1'600 * 2 = 3'200$	Ogni locale ha 2 finestre
Contiene	R	$2 * 4 * 30 * 4'800 = 1'152'000$	Il foraggio di ogni pasto prima di essere finito rimane nelle mangiatoie in media per 4 ore (ipotesi)
Controllo aria	R	$(6 * 30) * 1'600 = 288'000$	Un'occorrenza per ogni rilevazione dei parametri dei locali
Costituito	R	$10 * 40 = 400$	10 stalle per ogni agriturismo
Delimitazione	R	600	Ogni zona di pascolo ha una recinzione
Dispone	R	$15 * 40 = 600$	Ogni agriturismo dispone di 15 zone di pascolo
Divisa	R	$400 * 4 = 1'600$	Ogni stalla ha 4 locali
Interruzione	R	405	Un'occorrenza per ogni gestazione interrotta
Nascita	R	$8'085 * 0,95 = 7'680$	Il 95% delle gestazioni viene portato a termine
Necessario	R	$48'510 / 11 = 4'410$	1 controllo su 11 richiede un esame diagnostico aggiuntivo
Ospita	R	$1'600 * 12 = 19'200$	Ogni locale ospita 12 animali
Pascolo	R	$60 * 8 * 21 * 19'200 = 193'536'000$	Un'occorrenza per ogni volta che viene salvata la posizione durante il pascolo
Proprietà acqua	R	$12 * 15 * 9'600 = 1'728'000$	Mediamente per 15 giorni su 30 non ci sono sostanze disciolte nell'acqua degli abbeveratoi (ipotesi)
Responsabile	R	8'085	Ogni gestazione ha un veterinario responsabile
Richiesta	R	$10 * 1'600 = 16'000$	Un'occorrenza per ogni richiesta di pulizia dei locali

Sensore abbeveratoio	R	$12 * 30 * 9'600 = 3'456'000$	Un'occorrenza per ciascuna misurazione dei sensori negli abbeveratoi
Sensore mangiatoia	R	$2 * 6 * 30 * 4'800 = 1'728'000$	Un'occorrenza per ciascuna misurazione dei sensori nelle mangiatoie
Sottoposta	R	48'510	Un'occorrenza per ogni controllo diagnostico
Successo	R	$10'105 * 0,8 = 8'085$	L'80% delle riproduzioni ha successo
Supervisiona	R	10'105	Ogni riproduzione è supervisionata da un veterinario
Svolge	R	48'510	Ogni controllo è svolto da un veterinario
Tentativo	R	10'105	Un'occorrenza per ogni riproduzione

5.1.2) Area Produzione

Concetto	Tipo	Volume	Motivazione
Cantina	E	$15 * 40 = 600$	15 cantine per ogni agriturismo (ipotesi)
Caratteristica	E	$864'000 * 6 = 5'184'000$	Ogni unità di latte prodotta ha 6 sostanze chimiche all'interno
Fase	E	$25 * 4 = 100$	Ogni ricetta dei formaggi ha 4 fasi (ipotesi)
Formaggio	E	25	Ipotesi
Laboratorio produzione	E	$2 * 40 = 80$	Mediamente ci sono 2 laboratori per ogni agriturismo (ipotesi)
Latte	E	$(19'200 * 0,75) * 2 * 30 = 864'000$	Ogni volta che un animale di sesso femminile viene munto (2 volte al giorno) vengono salvate le informazioni sul latte, memorizzate per un mese (ipotesi)
Lotto	E	$80 * 500 = 40'000$	Ogni laboratorio ha prodotto mediamente 500 lotti (ipotesi)
Magazzino	E	$8 * 40 = 320$	8 magazzini per ogni agriturismo (ipotesi)
Mungitrice	E	$(19'200 * 0,75) / 16 = 900$	Il 75% degli animali sono di sesso femminile ed è necessaria 1 mungitrice ogni 16 femmine (ipotesi)

Operaio	E	1'300	Ipotesi
Parametro di processo	E	$100 * 5 = 500$	Ogni fase ha 5 parametri di processo (ipotesi)
Parametro effettivo	E	$4 * 5 = 20 * 1'400'000 = 28'000'000$	Ogni prodotto ha in media 4 fasi e ogni fase ha in media 5 parametri di processo
Prodotto caseario	E	$40'000 * 35 = 1'400'000$	Ogni lotto contiene 35 prodotti caseari (ipotesi)
Scaffalatura	E	$10 * 600 + 8 * (320 - 120) = 7'600$	10 scaffalature per ogni cantina e 8 per ogni magazzino, 120 dei quali però sono senza scaffali (ipotesi)
Silos	E	$2 * 40 = 80$	In media ci sono 2 silos per agriturismo (ipotesi)
Stato cantina	E	$600 * 30 = 18'000$	Una volta al giorno viene salvato lo stato di ogni cantina, memorizzato per un mese
Collocazione	R	1'400'000	Un'occorrenza per ogni formaggio prodotto
Contenuto	R	864'000	Un'occorrenza per ogni mungitura
Dotato	R	600	Un'occorrenza per ogni cantina
Formato	R	1'400'000	Un'occorrenza per ogni formaggio prodotto
Lavora	R	$8 * 40'000 = 320'000$	Su ogni lotto lavorano in media 8 operai (ipotesi)
Mungitura	R	864'000	Un'occorrenza per ogni volta che viene effettuata una mungitura
Processo produttivo	R	28'000'000	Un'occorrenza per ciascun parametro di processo di ogni prodotto caseario
Produzione	R	40'000	Un'occorrenza per ogni lotto prodotto
Proprietà	R	80	Un'occorrenza per ogni laboratorio
Proprietà latte	R	5'184'000	Un'occorrenza per ogni sostanza presente in ogni unità di latte estratta
Provvisto	R	320	Un'occorrenza per ogni magazzino
Ricetta	R	100	Un'occorrenza per ciascuna fase

Sensore cantina	R	18'000	Un'occorrenza per ogni salvataggio dello stato cantina
Stagionatura	R	$10 * 600 = 6'000$	Per ogni cantina ci sono 10 scaffalature
Stoccaggio	R	$(320 - 120) * 8 = 1'600$	8 scaffalature per ogni magazzino, 120 dei quali però sono senza scaffali
Tipologia	R	1'400'000	Un'occorrenza per ogni formaggio prodotto
Ubicazione	R	80	Un'occorrenza per ogni silos
Valore ideale	R	500	Un'occorrenza per ogni parametro di processo

5.1.3) Area Healthcare

Concetto	Tipo	Volume	Motivazione
Disturbo	E	$1'152'000 / 100 = 11'520$	Ogni 100 visite viene rilevato un disturbo (ipotesi)
Farmaco	E	200	Ipotesi
Indicatore benessere	E	13	Possibili indicatori oggettivi e soggettivi rilevabili
Lesione	E	$1'152'000 / 40 = 28'800$	Ogni 40 visite viene rilevata una lesione (ipotesi)
Orario somministrazione	E	$200 * 3 = 600$	In media per ogni farmaco ci sono 3 orari di somministrazione (ipotesi)
Patologia	E	25	Ipotesi
Terapia	E	$3'465 + (1'152'000 / 75) = 18'825$	Somma delle terapie prescritte dai controlli e dalle visite (1 ogni 75) (ipotesi)
Visita	E	$96 * 12'000 = 1'152'000$	Ogni veterinario ha visitato in media 12'000 animali (ipotesi)
Assunzione	R	600	Un'occorrenza per ogni orario di somministrazione
Compie	R	1'152'000	Un'occorrenza per ogni visita
Esordio	R	24'000	Un'occorrenza per ogni terapia prescritta da una visita
Monitoraggio	R	$(1'152'000 / 30) * 6 = 230'400$	In media ogni 30 visite vengono misurati 6 indici (ipotesi)
Rilevazione	R	11'520	Un'occorrenza per ogni disturbo
Soggetto	R	1'152'000	Un'occorrenza per ogni visita

Somministra	R	$18'825 * 3 = 56'475$	Per ogni terapia vengono prescritti in media 3 farmaci (ipotesi)
Valutazione	R	28'800	Un'occorrenza per ogni lesione

5.1.4) Area Soggiorno

Concetto	Tipo	Volume	Motivazione
Account	E	$40'000 * 0,75 = 30'000$	Il 75% dei clienti si registra sul sito web (ipotesi)
Area tenuta	E	$14 * 40 = 560$	14 aree per ogni agriturismo (ipotesi)
Camera	E	$30 * 40 = 1'200$	30 camere per agriturismo (ipotesi)
Cliente	E	40'000	Ipotesi
Escursione	E	$650 * 40 = 26'000$	Ci sono in media 650 escursioni in ogni agriturismo (ipotesi)
Guida	E	$3 * 40 = 120$	3 guide per ogni agriturismo (Ipotesi)
Letto	E	$2 * 1'200 = 2'400$	In media 2 letti per camera (ipotesi)
Pagamento	E	$1,6 * 20'000 = 32'000$	In media ci sono 1,6 pagamenti per ogni prenotazione (ipotesi)
Prenotazione	E	$40'000 * 0,5 = 20'000$	In media ogni cliente ha effettuato 0,5 prenotazioni (ipotesi)
Servizio aggiuntivo	E	5	Ipotesi
Step	E	$26'000 * 5 = 130'000$	Ogni escursione visita mediamente 5 aree (ipotesi)
Afferenza guida	R	120	Un'occorrenza per ogni guida
Affitto	R	1'200	Un'occorrenza per ogni camera
Assegnazione	R	$20'000 * 1,4 = 28'000$	Per ogni prenotazione vengono richieste mediamente 1,4 camere (ipotesi)
Associata	R	26'000	Un'occorrenza per ogni escursione
Capienza	R	2'400	Un'occorrenza per ogni letto
Effettua	R	20'000	Un'occorrenza per ogni prenotazione
Itinerario	R	130'000	Un'occorrenza per ogni step
Percorre	R	130'000	Un'occorrenza per ogni step
Prenota	R	$1,5 * 20'000 = 30'000$	In media vengono richieste 1,5 escursioni per ogni prenotazione (ipotesi)

Registrazione	R	30'000	Un'occorrenza per ogni account
Richiede	R	$20'000 * 0,6 = 12'000$	In ogni prenotazione sono richiesti in media 0,6 servizi aggiuntivi (ipotesi)
Situata	R	560	Un'occorrenza per ogni area della tenuta
Transazione	R	32'000	Un'occorrenza per ogni pagamento

5.1.5) Area Store

Concetto	Tipo	Volume	Motivazione
Centro smistamento	E	20	Ipotesi
Documento	E	$30'000 * 0,8 = 24'000$	Circa l'80% dei clienti registrati inserisce le informazioni riguardanti il documento (ipotesi)
Ordine	E	$30'000 * 0,75 * 5 = 112'500$	Il 75% dei clienti registrati acquista formaggi, in media per 5 volte (ipotesi)
Recensione	E	$(30'000 * 0,75) / 6 = 3'750$	1 utente su 6 lascia una recensione (ipotesi)
Scheda valutazione	E	$1'012'500 * 0,15 = 151'875$	Il 15% dei prodotti venduti viene restituito (ipotesi)
Spedizione	E	112'500	Una spedizione per ogni ordine
Attesa	R	$112'500 * 0,01 * 3 = 3'375$	Gli ordini pendenti sono l'1% di quelli totali e per ogni ordine vengono richiesti in media 3 tipi di formaggi (ipotesi)
Esegue	R	112'500	Un'occorrenza per ogni ordine
Fornitura	R	$112'500 * 9 = 1'012'500$	In ogni ordine vengono forniti in media 9 prodotti caseari (ipotesi)
Invio	R	112'500	Un'occorrenza per ogni spedizione
Percorso	R	$112'500 * 5 = 562'500$	Ogni spedizione passa in media da 5 hub (ipotesi)
Possiede	R	24'000	Un'occorrenza per ogni documento
Reso	R	151'875	Un'occorrenza per ogni scheda di valutazione
Riceve	R	3'750	Un'occorrenza per ogni recensione

5.2) Individuazione delle operazioni significative e tavola degli accessi

5.2.1) Camere da preparare

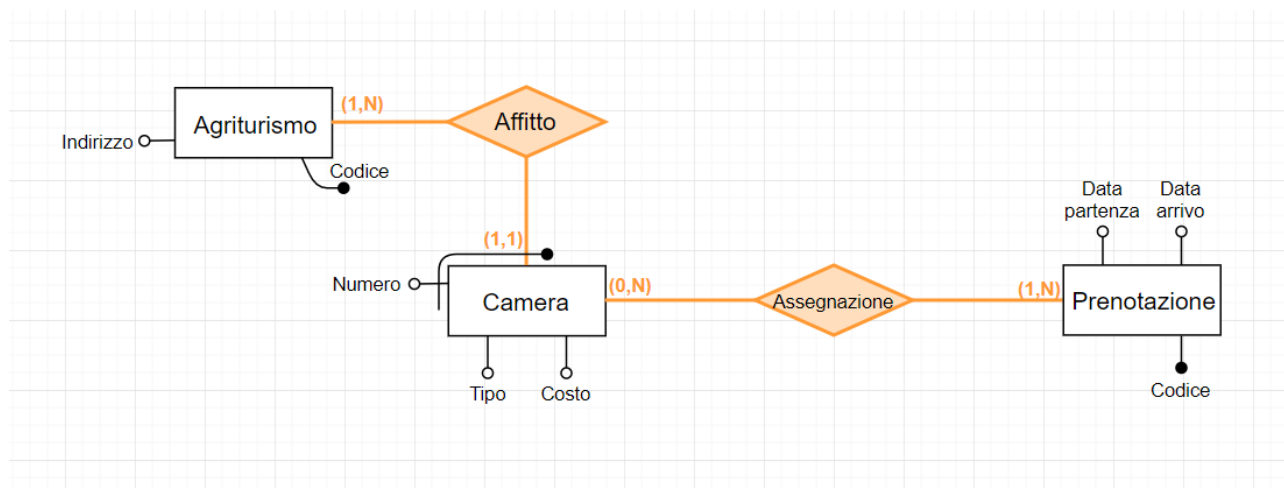
Descrizione: Dato l'agriturismo vengono controllate tutte le camere prenotate per il giorno successivo in modo che possano essere preparate per i nuovi ospiti in arrivo.

Input: Codice agriturismo

Output: Numero delle camere da preparare

Frequenza giornaliera: 1 volta al giorno per ogni agriturismo = 40 volte al giorno

Porzione di diagramma interessata:



Porzione tavola dei volumi interessata:

Affitto	Relazione	1.200
Agriturismo	Entità	40
Assegnazione	Relazione	28.000
Camera	Entità	1.200
Prenotazione	Entità	20.000

Tavola degli accessi:

Numero operazioni elementari	Tipo	Tipo costruito	Nome costruito	Descrizione
30	L	Relazione	Affitto	Otengo le camere dell'agriturismo
720 ¹	L	Relazione	Assegnazione	Otengo le camere prenotate
720	L	Entità	Prenotazione	Controllo la data di arrivo di ogni prenotazione
Totale operazioni elementari				1.470 ²
Totale operazioni elementari al giorno				58.800 ³

1. In media ogni camera è stata prenotata 24 volte = 30 * 24 = 720

2. Calcolo totale delle operazioni elementari per singola operazione:
 $30 + 720 + 720 = 1.470$
3. Calcolo totale delle operazioni elementari giornaliere:
 $1.470 * 40 = 58.800$

5.2.2) Stagionatura formaggi

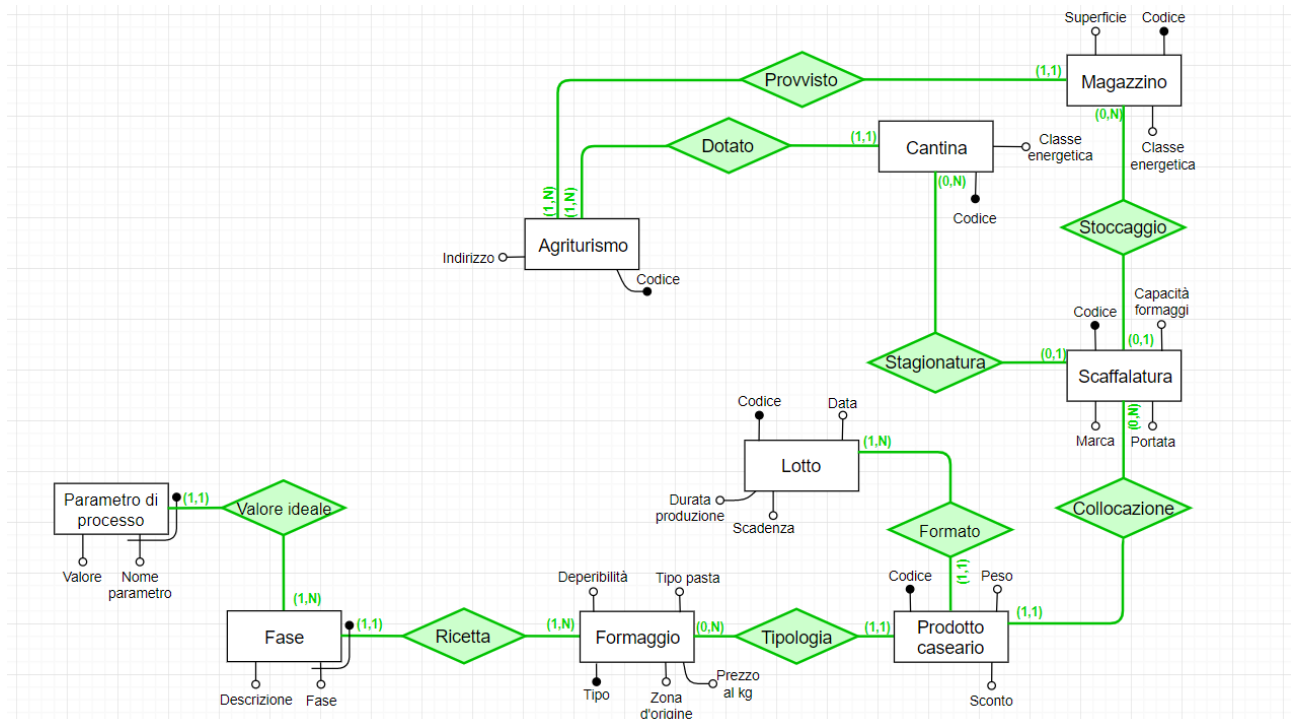
Descrizione: Ogni giorno viene controllato il processo di stagionatura dei formaggi cosicché, se terminato, essi vengano spostati nei magazzini (mantenendo, per quanto possibile, i prodotti spostati di uno stesso lotto sullo stesso scaffale per ottimizzare le operazioni di vendita).

Input: Nessuno

Output: Nessuno

Frequenza giornaliera: 1 volta al giorno

Porzione di diagramma interessata:



Porzione tavola dei volumi interessata:

Agriturismo	Entità	40
Cantina	Entità	600
Collocazione	Relazione	1.400.000
Dotato	Relazione	600
Fase	Entità	100
Formaggio	Entità	25
Formato	Relazione	1.400.000
Lotto	Entità	40.000
Magazzino	Entità	320
Parametro di processo	Entità	500
Prodotto caseario	Entità	1.400.000
Provvisto	Relazione	320

Ricetta	Relazione	100
Scaffalatura	Entità	7.600
Stagionatura	Relazione	6.000
Stoccaggio	Relazione	1.600
Tipologia	Relazione	1.400.000
Valore ideale	Relazione	500

Tavola degli accessi:

Numero operazioni elementari	Tipo	Tipo costruito	Nome costruito	Descrizione
6.000	L	Relazione	Stagionatura	Ottingo tutti gli scaffali presenti nelle cantine
140.000 ¹	L	Relazione	Collocazione	Ottingo i formaggi ancora collocati sulle scaffalature
140.000	L	Relazione	Formato	Trovo il lotto dei formaggi
4.000 ²	L	Entità	Lotto	Leggo la data di produzione (cioè quella in cui è iniziata la stagionatura)
140.000	L	Relazione	Tipologia	Ottingo la tipologia dei formaggi
100	L	Relazione	Ricetta	Ottingo tutte le fasi delle ricette di tutti i formaggi
500	L	Relazione	Valore ideale	Trovo tutti i parametri di processo
25	L	Entità	Parametro di processo	Leggo i tempi di stagionatura
1.600	L	Relazione	Stoccaggio	Prendo tutte le scaffalature dei magazzini
1.600	L	Relazione	Collocazione	Verifico quali scaffalature siano già piene
150 ³ * 2	S	Relazione	Collocazione	Se la stagionatura è finita sposto il prodotto in magazzino
Totale operazioni elementari	434.125 ⁴			
Totale operazioni elementari al giorno	434.125 ⁵			

1. Si stima che il 10% di tutti i prodotti caseari sia in stagionatura = $1.400.000 * 0,1 = 140.000$
2. Ogni lotto contiene in media 35 prodotti = $140.000 / 35 = 4.000$
3. In media 150 formaggi al giorno terminano la stagionatura

4. Calcolo totale delle operazioni elementari per singola operazione:
 $6.000 + 140.000 + 140.000 + 4.000 + 140.000 + 100 + 500 + 25 + 1.600 + 1.600 + 300 = 434.125$
5. Calcolo totale delle operazioni elementari giornaliere:
 $434.125 * 1 = 434.125$

5.2.3) Report vendite

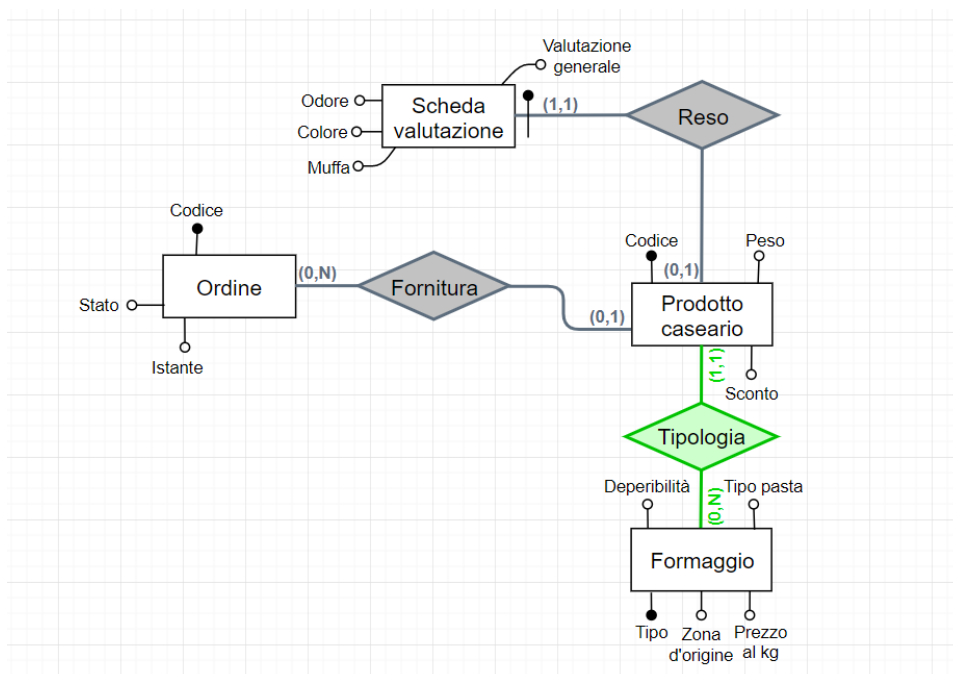
Descrizione: Data la tipologia di un formaggio vengono calcolate le vendite e i resi relativi agli ordini eseguiti nell'arco di una settimana

Input: Tipologia formaggio

Output: Vendite e resi settimanali

Frequenza giornaliera: L'operazione è eseguita 1 volta alla settimana, quindi $1/7 = 0,15 * 25$ (numero tipologie di formaggio) = 3,75

Porzione di diagramma interessata:



Porzione tavola dei volumi interessata:

Formaggio	Entità	25
Fornitura	Relazione	1.012.500
Ordine	Entità	112.500
Prodotto caseario	Entità	1.400.000
Tipologia	Relazione	1.400.000

Tavola degli accessi:

Numero operazioni elementari	Tipo	Tipo costruito	Nome costruito	Descrizione
56.000 ¹	L	Relazione	Tipologia	Trovo i prodotti di quella tipologia di formaggi

47.600 ²	L	Relazione	Fornitura	Conto i formaggi venduti
7.140 ³	L	Relazione	Resi	Conto i formaggi resi
Totale operazioni elementari		103.600 ⁴		
Totale operazioni elementari al giorno		415.275 ⁵		

1. $1.400.000$ (prodotti caseari) / 25 (numero formaggi) = 56.000
2. $56.000 - 8.400$ (si stima che il 15% dei prodotti non sia ancora stato venduto) = 47.600
3. Si stima che solo il 15% dei prodotti venga restituito
4. Calcolo totale delle operazioni elementari per singola operazione:
 $56.000 + 47.600 + 7.140 = 110.740$
5. Calcolo totale delle operazioni elementari giornaliere:
 $103.600 * 3,75 = 415.275$

5.2.4) Rilevamento posizione

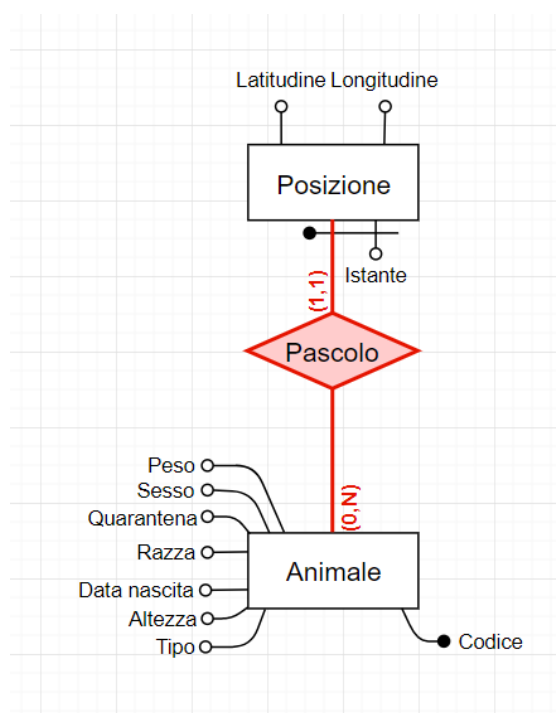
Descrizione: Date le informazioni rilevate da un sensore GPS durante il pascolo, si memorizza la posizione dell'animale.

Input: Codice animale, latitudine, longitudine

Output: Nessuno

Frequenza giornaliera: Ogni minuto per le otto ore di pascolo di ogni animale = $60 * 8 * 19.200 = 9.216.000$

Porzione di diagramma interessata:



Porzione tavola dei volumi interessata:

Animale	Entità	19.200
Pascolo	Relazione	193.536.000

Posizione	Entità	193.536.000
-----------	--------	-------------

Tavola degli accessi:

Numero operazioni elementari	Tipo	Tipo costruito	Nome costruito	Descrizione
1 * 2	S	Entità	Posizione	Aggiungo una nuova posizione
1 * 2	S	Relazione	Pascolo	Collego la posizione all'animale
Totale operazioni elementari		4 ¹		
Totale operazioni elementari al giorno		36.864.000 ²		

1. Calcolo totale delle operazioni elementari per singola operazione:
 $2 + 2 = 4$
2. Calcolo totale delle operazioni elementari giornaliere:
 $4 * 9.216.000 = 36.864.000$

5.2.5) Saldo soggiorno

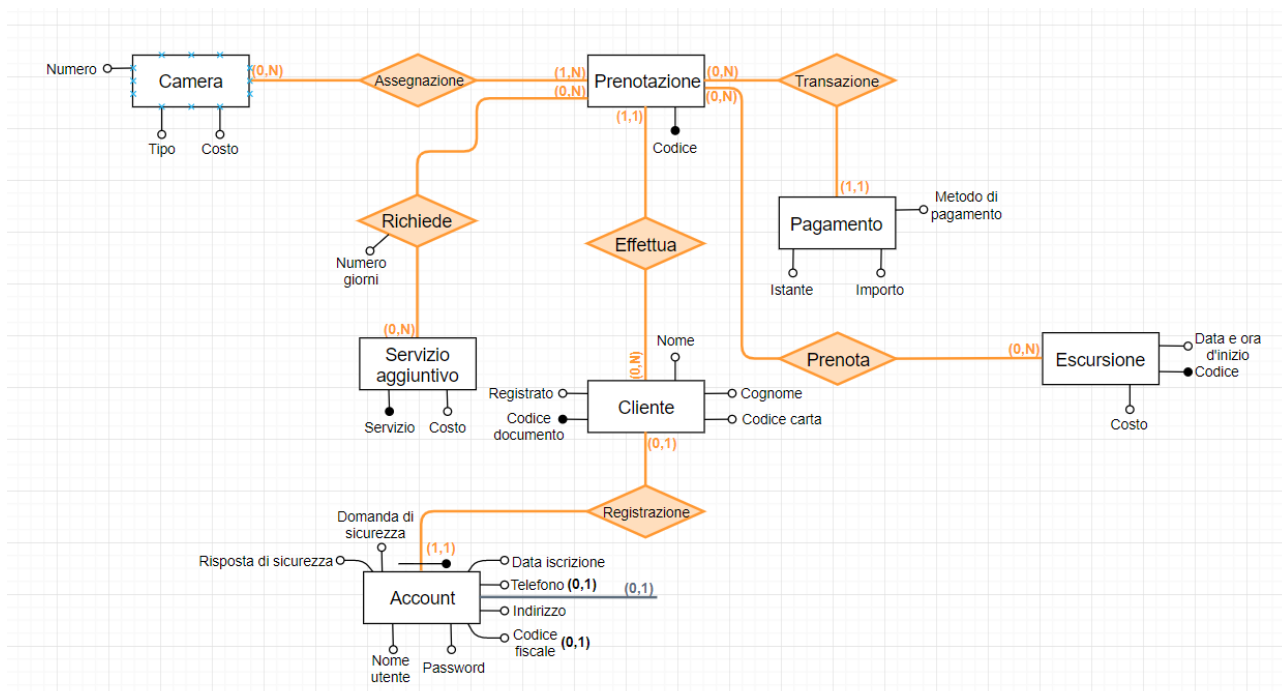
Descrizione: Al termine del soggiorno, dato il codice del cliente, viene trovata la sua prenotazione e viene calcolato il prezzo totale da pagare.

Input: Codice documento

Output: Somma da pagare

Frequenza giornaliera: 20.000 (prenotazioni in cinque anni) / 1.825 (giorni in 5 anni) = 11 (approssimato per eccesso)

Porzione di diagramma interessata:



Porzione tavola dei volumi interessata:

Account	Entità	30.000
Assegnazione	Relazione	28.000
Camera	Entità	1.200
Cliente	Entità	40.000
Effettua	Relazione	20.000
Escursione	Entità	26.000
Prenota	Relazione	30.000
Prenotazione	Entità	20.000
Registrazione	Relazione	30.000
Richiede	Relazione	12.000
Servizio aggiuntivo	Entità	5

Tavola degli accessi:

Numero operazioni elementari	Tipo	Tipo costruito	Nome costruito	Descrizione
1 ¹	L	Relazione	Effettua	Trovo le prenotazioni del cliente
1	L	Entità	Prenotazione	Controllo la data di partenza e arrivo
2 ²	L	Relazione	Assegnazione	Trovo tutte le camere assegnate
2	L	Entità	Camera	Leggo il costo delle camere
1 ³	L	Relazione	Richiede	Trovo gli eventuali servizi aggiuntivi e il numero di giorni per cui sono stati richiesti
1	L	Entità	Servizio aggiuntivo	Leggo il costo dei servizi
2 ⁴	L	Relazione	Prenota	Trovo tutte le escursioni prenotate
2	L	Entità	Escursione	Leggo il costo delle escursioni
1	L	Relazione	Transazione	Ottingo i pagamenti effettuati
1	L	Entità	Pagamento	Leggo l'importo già versato, se presente
Totale operazioni elementari		14 ⁵		
Totale operazioni elementari al giorno		154 ⁶		

1. Mediamente ogni cliente effettua 0,5 prenotazioni (approssimate per eccesso = 1)

2. In ogni prenotazione sono richieste in media 1,4 camere (approssimate per eccesso = 2)
3. In ogni prenotazione sono richiesti in media 0,6 servizi aggiuntivi (approssimati per eccesso = 1)
4. In media sono richieste 1,5 escursioni per prenotazione (approssimate per eccesso = 2)
5. Calcolo totale delle operazioni elementari per singola operazione:
 $1 + 1 + 2 + 2 + 1 + 1 + 2 + 2 + 1 + 1 = 14$
6. Calcolo totale delle operazioni elementari giornaliere:
 $14 * 11 = 154$

5.2.6) Aggiunta prodotto ad un ordine pendente

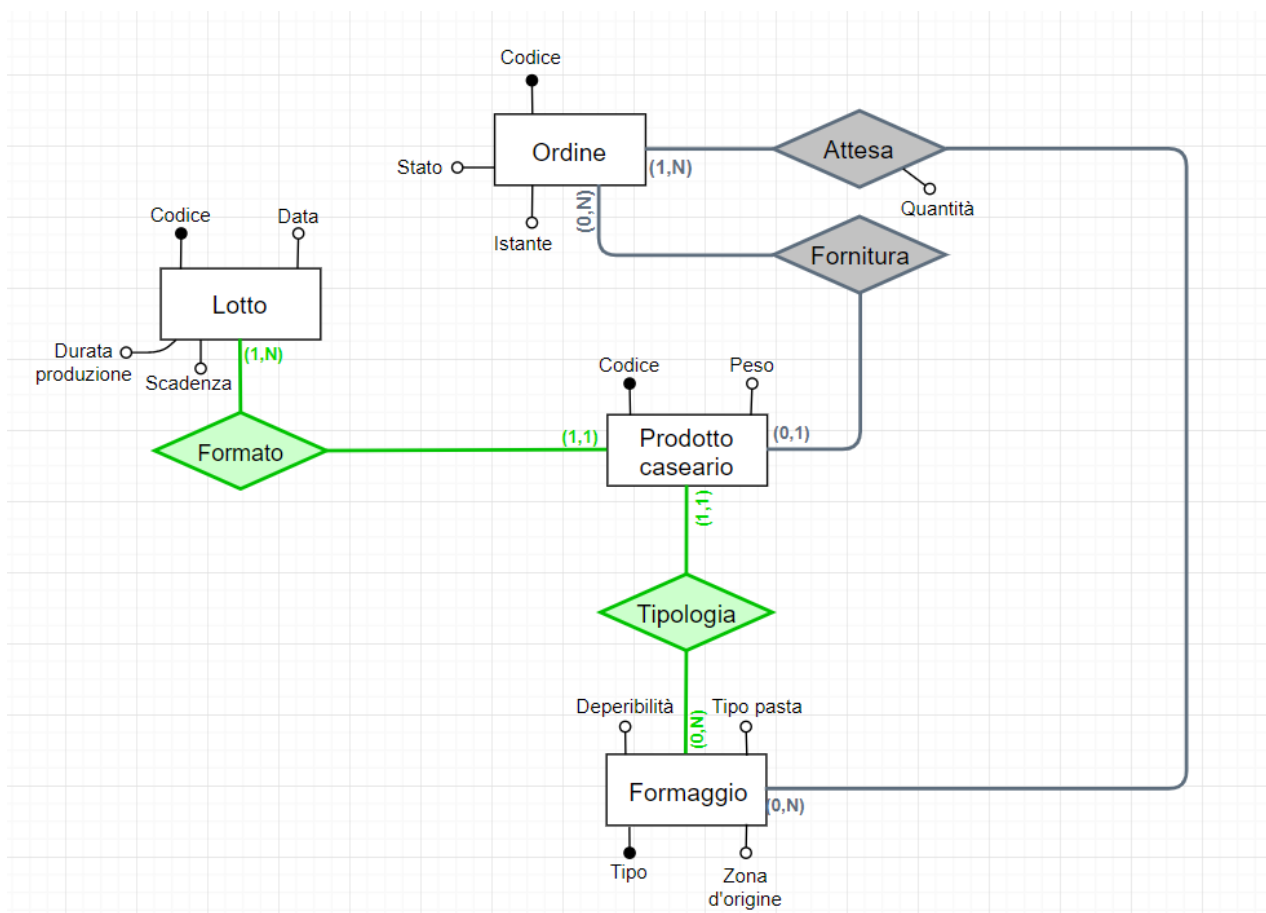
Descrizione: Ogni volta che viene prodotto un formaggio o è terminato un processo di stagionatura viene controllato se il prodotto caseario in questione è presente in un ordine pendente. Se così fosse, verrebbe aggiunto in coda alla spedizione che partirà quando tutto l'ordine sarà pronto.

Input: Codice prodotto caseario

Output: Nessuno

Frequenza giornaliera: $(1.400.000 \text{ (formaggi prodotti in 5 anni)} + 140.000 \text{ (prodotti stagionati in 5 anni)}) / 1.825 \text{ (giorni in 5 anni)} = 844 \text{ (approssimato per eccesso)}$

Porzione di diagramma interessata:



Porzione tavola dei volumi interessata:

Attesa	Relazione	3.375
Formaggio	Entità	25

Formato	Relazione	1.400.000
Fornitura	Relazione	1.012.500
Lotto	Entità	40.000
Ordine	Entità	112.500
Prodotto caseario	Entità	1.400.000
Tipologia	Relazione	1.400.000

Tavola degli accessi:

Numero operazioni elementari	Tipo	Tipo costruito	Nome costruito	Descrizione
1	L	Relazione	Fornitura	Verifico se il prodotto è già in una coda di spedizione
1	L	Relazione	Tipologia	Trovo la tipologia del formaggio
1	L	Entità	Prodotto caseario	Ottingo il peso del prodotto caseario
1.125 ¹	L	Relazione	Attesa	Trovo gli ordini pendenti che includono la tipologia di formaggio
1 * 2	S	Relazione	Fornitura	Aggiungo il formaggio se corrisponde alla quantità mancante
1 * 2	S	Entità	Ordine	Se l'ordine è pronto imposto lo stato "in processazione"
Totale operazioni elementari	1.132 ²			
Totale operazioni elementari al giorno	955.408 ³			

1. Gli ordini pendenti sono l'1% dei totali = $112.500 * 0,01 = 1.125$. Dato che, nel caso peggiore, la tipologia di formaggio che sto considerando sarà presente in tutti gli ordini pendenti, li considero tutti
2. Calcolo totale delle operazioni elementari per singola operazione:
 $1 + 1 + 1 + 1.125 + 2 + 2 = 1.132$
3. Calcolo totale delle operazioni elementari giornaliere:
 $1.132 * 844 = 955.408$

5.2.7) Tentativo di riproduzione

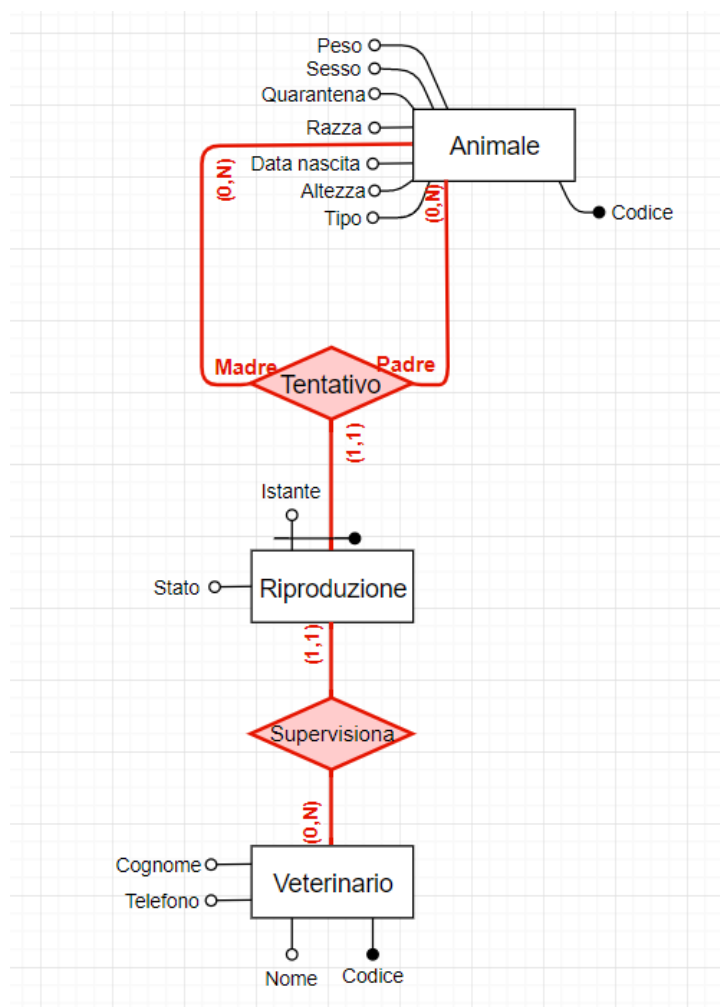
Descrizione: Dati due animali di sesso opposto e il veterinario supervisore viene aggiunto un tentativo di riproduzione.

Input: Codice madre, codice padre, codice veterinario

Output: Nessuno

Frequenza giornaliera: $10.105 \text{ (riproduzioni in cinque anni)} / 1.825 \text{ (giorni in cinque anni)} = 6$ (approssimato per eccesso)

Porzione di diagramma interessata:



Porzione tavola dei volumi interessata:

Animale	Entità	19.200
Tentativo	Relazione	10.105
Riproduzione	Entità	10.105
Supervisiona	Relazione	10.105
Veterinario	Entità	96

Tavola degli accessi:

Numero operazioni elementari	Tipo	Tipo costruito	Nome costruito	Descrizione
------------------------------	------	----------------	----------------	-------------

1 * 2	S	Entità	Riproduzione	Creo la nuova riproduzione
1 * 2	S	Relazione	Tentativo	Collego il padre e la madre alla riproduzione
1 * 2	S	Relazione	Supervisiona	Collego la riproduzione al veterinario responsabile
Totale operazioni elementari		6 ¹		
Totale operazioni elementari al giorno		30 ²		

1. Calcolo totale delle operazioni elementari per singola operazione:
2 + 2 + 2 = 6
2. Calcolo totale delle operazioni elementari giornaliere:
6 * 6 = 30

5.2.8) Animali sotto terapie

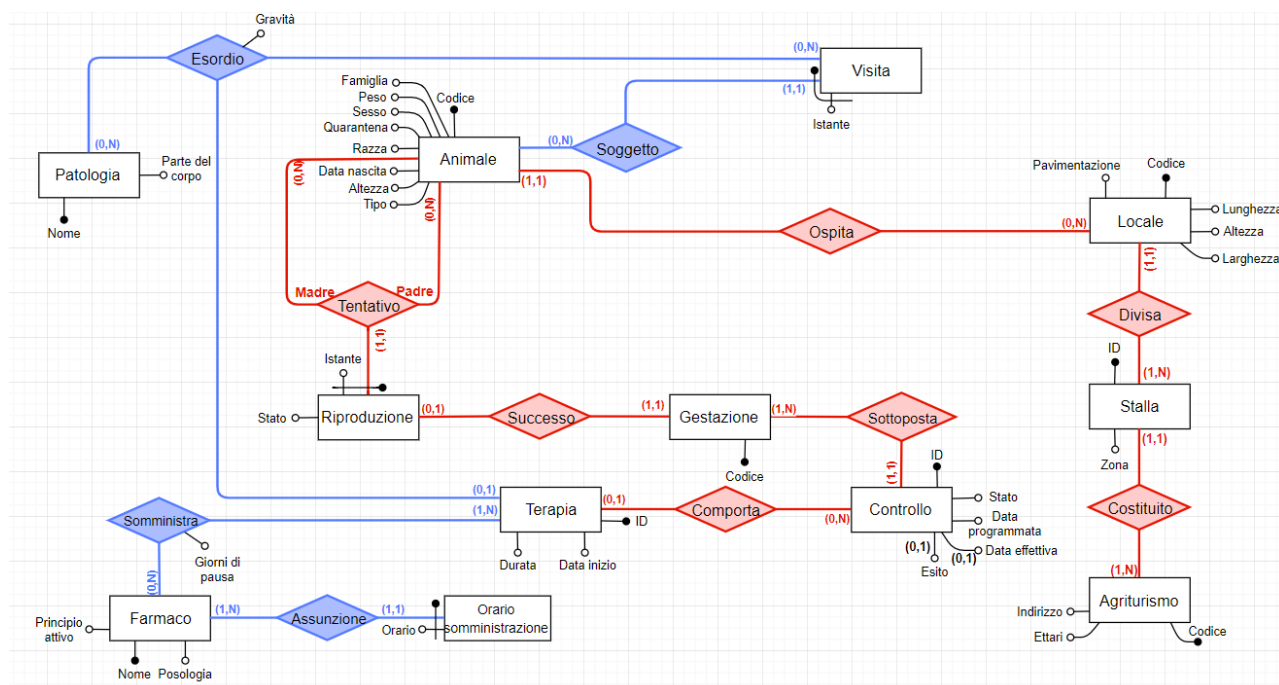
Descrizione: Ogni giorno, in tutti gli agriturismi, vengono ottenuti gli animali attualmente sottoposti a terapie.

Input: Codice agriturismo

Output: Animali di un agriturismo in cura

Frequenza giornaliera: 1 volta al giorno per ogni agriturismo = 40 volte al giorno

Porzione di diagramma interessata:



Porzione tavola dei volumi interessata:

Agriturismo	Entità	40
Animale	Entità	19.200
Comporta	Relazione	3.465

Controllo	Entità	48.510
Costituito	Relazione	400
Divisa	Relazione	1.600
Esordio	Relazione	24.000
Gestazione	Entità	8.085
Locale	Entità	1.600
Ospita	Relazione	19.200
Patologia	Entità	25
Riproduzione	Entità	10.105
Soggetto	Relazione	1.152.000
Sottoposta	Relazione	48.510
Stalla	Entità	400
Successo	Relazione	8.085
Tentativo	Relazione	10.105
Terapia	Entità	18.825
Visita	Entità	1.152.000

Tavola degli accessi:

Numero operazioni elementari	Tipo	Tipo costruito	Nome costruito	Descrizione
18.825	L	Entità	Terapia	Controllo quali terapie sono in corso
1.506 ¹	L	Relazione	Comporta	Trovo i controlli che hanno prescritto le terapie
251 ²	L	Relazione	Sottoposta	Ottingo le gestazioni relative ai controlli
251	L	Relazione	Successo	Ottingo i tentativi di riproduzione
251	L	Relazione	Tentativo	Ottingo gli animali di sesso femminile sottoposti a terapie
377 ³	L	Relazione	Esordio	Ottingo le visite che hanno prescritto le terapie
377	L	Relazione	Soggetto	Ottingo gli animali oggetto delle visite
628 ⁴	L	Relazione	Ospita	Trovo i locali a cui appartengono gli animali
53 ⁵	L	Relazione	Divisa	Trovo le stalle a cui appartengono i locali

14 ⁶	L	Relazione	Costituito	Controllo se l'agriturismo a cui appartengono le stalle è quello corretto
Totale operazioni elementari		22.533 ⁷		
Totale operazioni elementari al giorno		1.315.120 ⁸		

1. Si stima che le terapie non concluse siano il 10% di quelle totali. 4/5 di queste sono prescritte a seguito di controlli = $(18.825 * 0,1) * 4 / 5 = 1.506$
2. Per ogni gestazione ci sono in media 6 controlli = $1.506 / 6 = 251$
3. 1.883 (terapie non concluse) / $5 = 377$ (visite che hanno prescritto terapie)
4. 251 (gestanti sottoposte a terapie) + 377 (animali sottoposti a terapie) = 628
5. Ogni locale ospita in media 12 animali = $628 / 12 = 53$
6. Ogni stalla contiene mediamente 4 locali = $53 / 4 = 14$
7. Calcolo totale delle operazioni elementari per singola operazione:
 $18.825 + 1.506 + 251 + 251 + 251 + 377 + 377 + 628 + 53 + 14 = 22.533$
8. Calcolo totale delle operazioni elementari giornaliere:
 $22.533 * 40 = 901.320$

5.2.9) Inserimento visita

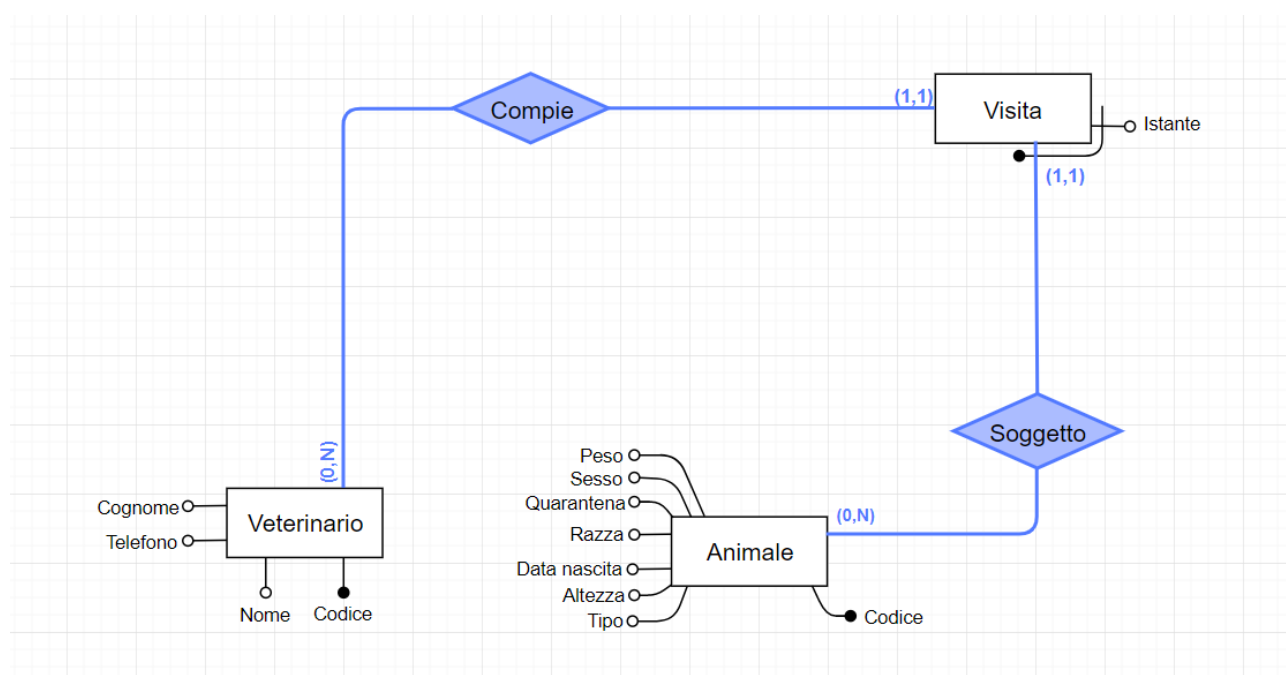
Descrizione: Appena eseguita la visita viene effettuato il relativo inserimento e viene aggiornata la scheda dell'animale.

Input: Codice animale, altezza, peso, codice veterinario

Output: Nessuno

Frequenza giornaliera: $1.152.000$ (visite in cinque anni) / 1.825 (giorni in cinque anni) = 632 (approssimato per eccesso)

Porzione di diagramma interessata:



Porzione tavola dei volumi interessata:

Animale	Entità	19.200
Compie	Relazione	1.152.000
Soggetto	Relazione	1.152.000
Veterinario	Entità	96
Visita	Entità	1.152.000

Tavola degli accessi:

Numero operazioni elementari	Tipo	Tipo costruito	Nome costruito	Descrizione
1 * 2	S	Entità	Visita	Creo la nuova visita
1 * 2	S	Relazione	Soggetto	Collego la visita all'animale
1 * 2	S	Relazione	Compie	Collego la visita al veterinario che l'ha eseguita
1 * 2	S	Entità	Animale	Aggiorno l'altezza e il peso dell'animale
Totale operazioni elementari		8 ¹		
Totale operazioni elementari al giorno		5.056 ²		

1. Calcolo totale delle operazioni elementari per singola operazione:
 $2 + 2 + 2 + 2 = 8$
2. Calcolo totale delle operazioni elementari giornaliere:
 $8 * 632 = 5.056$

5.3) Introduzione di ridondanze

Nel seguente paragrafo vengono analizzate alcune delle operazioni significative scelte per capire se possono beneficiare dall'introduzione di ridondanze in termini di mole di dati da processare. A seguito di una valutazione degli accessi sono state scelte le seguenti operazioni: Camere da preparare (5.2.1), Stagionatura formaggi (5.2.2) e Animali sotto terapie (5.2.8).

5.3.1) Camere da preparare

Viene aggiunto l'attributo **Data arrivo** sull'entità Camera.

Numero operazioni elementari	Tipo	Tipo costruito	Nome costruito	Descrizione
30	L	Relazione	Affitto	Ottengo le camere dell'agriturismo
30	L	Entità	Camera	Controllo la data di arrivo

Totale operazioni elementari	60 ¹
Totale operazioni elementari al giorno	2.400 ²

1. Calcolo totale delle operazioni elementari per singola operazione:
 $30 + 30 = 60$
2. Calcolo totale delle operazioni elementari giornaliere:
 $60 * 40 = 2.400$

Aggiornamento della ridondanza

L'attributo introdotto deve essere aggiornato ogni volta che le camere vengono assegnate.

Tavola degli accessi

Numero operazioni elementari	Tipo	Tipo costruito	Nome costruito	Descrizione
1	L	Entità	Prenotazione	Leggo la data di arrivo
1 * 2	S	Entità	Camera	Imposto la nuova data di arrivo
Totale operazioni elementari	3			
Totale operazioni elementari al giorno	48 ¹			

1. 28.000 (assegnazioni in 5 anni) / 1.825 (giorni in 5 anni) = 16 (assegnazioni giornaliere) * 3 = 48

Costi operazione

$$f^t = 40$$

$$o^t = 1.470$$

$$n^t = f^t * o^t = 40 * 1.470 = 58.800$$

$$o_{RID}^t = 60$$

$$n_{RID}^t = f^t * o_{RID}^t = 40 * 60 = 2.400$$

$$\Delta_{read} = n^t - n_{RID}^t = 58.800 - 2.400 = 56.400$$

$$g^A = 16$$

$$o^A = 3$$

$$n^A = g^A * o^A = 16 * 3 = 48$$

Come si può osservare dall'analisi costi-benefici $n^A < \Delta_{read}$, per cui è stata mantenuta la ridondanza.

5.3.2) Stagionatura formaggi

Viene aggiunto l'attributo **Fine stagionatura** sull'entità Prodotto Caseario.

Numero operazioni elementari	Tipo	Tipo costruito	Nome costruito	Descrizione
------------------------------	------	----------------	----------------	-------------

6.000	L	Relazione	Stagionatura	Ottingo tutti gli scaffali presenti nelle cantine
140.000 ¹	L	Relazione	Collocazione	Ottingo i formaggi ancora collocati sulle scaffalature
140.000	L	Entità	Prodotto caseario	Controllo se la stagionatura è terminata oggi
1.600	L	Relazione	Stoccaggio	Prendo le scaffalature dei magazzini
1.600	L	Relazione	Collocazione	Verifico quali scaffalature siano già piene
150 ² * 2	S	Relazione	Collocazione	Se la stagionatura è finita sposto il prodotto in magazzino
Totale operazioni elementari		289.500 ³		
Totale operazioni elementari al giorno		289.500 ⁴		

1. Si stima che il 10% di tutti i prodotti caseari sia in stagionatura = $1.400.000 * 0,1 = 140.000$
2. In media 150 formaggi al giorno terminano la stagionatura
3. Calcolo totale delle operazioni elementari per singola operazione:
 $6.000 + 140.000 + 140.000 + 1.600 + 1.600 + 300 = 289.500$
4. Calcolo totale delle operazioni elementari giornaliere:
 $289.500 * 1 = 289.500$

Aggiornamento della ridondanza

L'attributo introdotto deve essere soltanto inserito ogni volta che viene prodotto un formaggio.

Tavola degli accessi

Numero operazioni elementari	Tipo	Tipo costruito	Nome costruito	Descrizione
1	L	Relazione	Tipologia	Ottingo la tipologia del formaggio
4 ¹	L	Relazione	Ricetta	Ottingo le fasi della ricetta
20 ²	L	Relazione	Valore ideale	Trovo tutti i parametri di processo
1	L	Entità	Parametro di processo	Leggo la durata della stagionatura
1 * 2	S	Entità	Prodotto caseario	Scrivo la data di fine stagionatura
Totale operazioni elementari		28		

Totale operazioni elementari al giorno	21.504 ³
-----------------------------------------------	---------------------

1. Ogni formaggio, nella propria ricetta, ha 4 fasi
2. Ogni fase ha 5 valori ideali = $4 * 5 = 20$
3. Calcolo totale delle operazioni elementari giornaliere:
 $1.400.000$ (formaggi prodotti in 5 anni) / 1.825 (giorni in 5 anni) = 768 (formaggi prodotti giornalmente) * $28 = 21.504$

Costi operazione

$$f^t = 1$$

$$o^t = 434.125$$

$$n^t = f^t * o^t = 434.125$$

$$o_{RID}^t = 289.500$$

$$n_{RID}^t = f^t * o_{RID}^t = 289.500$$

$$\Delta_{read} = n^t - n_{RID}^t = 434.125 - 289.500 = 144.625$$

$$g^A = 768$$

$$o^A = 28$$

$$n^A = g^A * o^A = 768 * 28 = 21.504$$

Come si può osservare dall'analisi costi-benefici $n^A < \Delta_{read}$, per cui è stata mantenuta la ridondanza.

5.3.3) Animali sotto terapie

Viene aggiunta la relazione **In cura** che collega le entità Animale e Terapia.

Numero operazioni elementari	Tipo	Tipo costruito	Nome costruito	Descrizione
18.825	L	Entità	Terapia	Controllo quali terapie sono in corso
628 ¹	L	Relazione	In cura	Ottengo gli animali attualmente sottoposti a terapie
628	L	Relazione	Ospita	Trovo i locali a cui appartengono gli animali
53 ²	L	Relazione	Divisa	Trovo le stalle a cui appartengono i locali
14 ³	L	Relazione	Costituito	Controllo se l'agriturismo a cui appartengono le stalle è quello corretto
Totale operazioni elementari			20.148 ⁴	
Totale operazioni elementari al giorno			805.920 ⁵	

1. Animali attualmente sottoposti a terapie
2. Ogni locale ospita in media 12 animali = $628 / 12 = 53$
3. Ogni stalla contiene mediamente 4 locali = $53 / 4 = 14$

4. Calcolo totale delle operazioni elementari per singola operazione:
 $18.825 + 628 + 628 + 53 + 14 = 20.148$
5. Calcolo totale delle operazioni elementari giornaliere:
 $20.148 * 40 = 805.920$

Aggiornamento della ridondanza

La relazione introdotta deve essere soltanto inserita contestualmente all'aggiunta di una nuova terapia.

Tavola degli accessi

Numero operazioni elementari	Tipo	Tipo costruito	Nome costruito	Descrizione
1 * 2	S	Relazione	In cura	Collego la terapia all'animale
Totale operazioni elementari		2		
Totale operazioni elementari al giorno		22 ¹		

1. 18.825 (terapie prescritte in 5 anni) / 1.825 (giorni in 5 anni) = 11 (terapie prescritte giornalmente) * 2 = 22

Costi operazione

$$f^t = 40$$

$$o^t = 32.878$$

$$n^t = f^t * o^t = 40 * 22.533 = 901.320$$

$$o_{RID}^t = 27.765$$

$$n_{RID}^t = f^t * o_{RID}^t = 40 * 20.148 = 805.920$$

$$\Delta_{read} = n^t - n_{RID}^t = 901.320 - 805.920 = 95.400$$

$$g^A = 11$$

$$o^A = 2$$

$$n^A = g^A * o^A = 11 * 2 = 22$$

Come si può osservare dall'analisi costi-benefici $n^A < \Delta_{read}$, per cui è stata mantenuta la ridondanza.

5.4) Operazioni di impatto sulle ridondanze

Di seguito vengono analizzate le operazioni di lettura e scrittura che vanno ad impattare con le ridondanze introdotte nel paragrafo precedente.

5.4.1) Assegnazione camere

Descrizione: A seguito di una prenotazione da parte di un cliente vengono assegnate le camere.

Input: Codice prenotazione, numero camera, codice agriturismo

Output: Nessuno

Frequenza giornaliera: 28.000 (assegnazioni in cinque anni) / 1.825 (giorni in cinque anni) = 16 (arrotondato per eccesso)

Tavola degli accessi:

Numero operazioni elementari	Tipo	Tipo costruito	Nome costruito	Descrizione
$1 * 2$	S	Relazione	Assegnazione	Assegno la prenotazione alla camera
1	L	Entità	Prenotazione	Leggo la data di arrivo
$1 * 2$	S	Entità	Camera	Aggiorno la ridondanza
Totale operazioni elementari		5		
Totale operazioni elementari al giorno		80^1		

1. Calcolo totale delle operazioni elementari giornaliere:
 $5 * 16 = 80$

5.4.2) Lettura camera

Descrizione: Data una camera viene controllato fino a che giorno è libera.

Input: Numero camera, codice agriturismo

Output: Data a partire dalla quale la camera è libera

Frequenza giornaliera: 10

Tavola degli accessi:

Numero operazioni elementari	Tipo	Tipo costruito	Nome costruito	Descrizione
1	L	Entità	Camera	Leggo la data di arrivo ¹
24^2	L	Relazione	Assegnazione	Ottingo le prenotazioni
24	L	Entità	Prenotazione	Leggo la data di partenza corretta
Totale operazioni elementari		49		
Totale operazioni elementari al giorno		490^3		

1. Nel caso in cui la data sia già passata potrebbe indicare che la camera è libera così come che il cliente è già arrivato, quindi dovrò leggere anche la data di partenza
2. In media ogni camera è stata prenotata 24 volte
3. Calcolo totale delle operazioni elementari giornaliere:
 $49 * 10 = 490$

5.4.3) Inserimento prodotto caseario

Descrizione: Inserisco un nuovo prodotto caseario appena prodotto.

Input: Tipo formaggio, codice lotto, codice scaffalatura, peso

Output: Nessuno

Frequenza giornaliera: $1.400.000$ (formaggi prodotti in 5 anni) / 1.825 (giorni in 5 anni) = 768

Tavola degli accessi:

Numero operazioni elementari	Tipo	Tipo costruito	Nome costruito	Descrizione
$1 * 2$	S	Entità	Prodotto caseario	Inserisco il nuovo prodotto caseario
$1 * 2$	S	Relazione	Tipologia	Collego la tipologia di formaggio al prodotto
$1 * 2$	S	Relazione	Formato	Collego il lotto al prodotto caseario
$1 * 2$	S	Relazione	Collocazione	Collego il prodotto alla scaffalatura
4^1	L	Relazione	Ricetta	Ottingo le fasi della ricetta
20^2	L	Relazione	Valore ideale	Trovo tutti i parametri di processo
1	L	Entità	Parametro di processo	Leggo la durata della stagionatura
$1 * 2$	S	Entità	Prodotto caseario	Scrivo la data di fine stagionatura
Totale operazioni elementari		33		
Totale operazioni elementari al giorno		25.344^3		

1. Ogni formaggio, nella propria ricetta, ha 4 fasi
2. Ogni fase ha 5 valori ideali = $4 * 5 = 20$
3. Calcolo totale delle operazioni elementari giornaliere:
 $33 * 768 = 25.344$

5.4.4) Aggiornamento scadenza

Descrizione: Aggiorna la scadenza in base alla data di fine stagionatura.

Input: Codice prodotto caseario

Output: Nessuno

Frequenza giornaliera: $1.400.000$ (formaggi prodotti in 5 anni) / 1.825 (giorni in 5 anni) = 768

Tavola degli accessi:

Numero operazioni elementari	Tipo	Tipo costruito	Nome costruito	Descrizione
1	L	Entità	Prodotto caseario	Letto la data di fine stagionatura
1	L	Relazione	Formato	Ottingo il lotto
1 * 2	S	Entità	Lotto	Aggiorno la scadenza
Totale operazioni elementari		4		
Totale operazioni elementari al giorno		3.072 ¹		

1. Calcolo totale delle operazioni elementari giornaliere:

$$4 * 768 = 3.072$$

5.4.5) Inserimento terapia

Descrizione: Inserisco una nuova terapia appena prescritta.

Input: Data inizio, durata, animale

Output: Nessuno

Frequenza giornaliera: 18.825 (terapie in 5 anni) / 1.825 (giorni in 5 anni) = 11

Tavola degli accessi:

Numero operazioni elementari	Tipo	Tipo costruito	Nome costruito	Descrizione
1 * 2	S	Entità	Terapia	Creo la nuova terapia
1 * 2	S	Entità	In cura	Collego la terapia all'animale
Totale operazioni elementari		4		
Totale operazioni elementari al giorno		44		

5.4.6) Storico terapie

Descrizione: Dato un animale vengono trovate tutte le terapie a cui è stato sottoposto.

Input: Codice animale

Output: Terapie a cui è stato sottoposto

Frequenza giornaliera: 8

Tavola degli accessi:

Numero operazioni elementari	Tipo	Tipo costruito	Nome costruito	Descrizione
1 ¹	L	Relazione	In cura	Ottingo tutte le terapie dell'animale
Totale operazioni elementari		1		
Totale operazioni elementari al giorno		8		

1. 18.825 (terapie totali) / 19.200 (numero animali) = 1 (approssimato per eccesso)

6. Traduzione verso il modello relazionale

Avendo ottenuto il diagramma E-R ristrutturato e corretto si procede quindi alla traduzione nel modello logico relazionale; a differenza delle sezioni precedenti non è stata effettuata una suddivisione in aree tematiche in quanto entità di aree diverse possono essere messe in relazione tramite associazioni. Le scelte effettuate nella traduzione, quindi l'accorpamento delle relazioni in entità, sono state prese seguendo le regole spiegate durante il corso ad eccezione di alcune situazioni in cui sono state studiate caso per caso, al fine di ottimizzare i costi pur garantendo chiarezza ed efficienza; in questo capitolo vengono inoltre mostrati i vari vincoli di integrità referenziale e generici.

6.1) Modello relazionale

Abbeveratoio(Codice, Capienza, Locale)

Account(CodiceDocumento, Username, Password, CodiceFiscale, Indirizzo, Telefono, DataIscrizione, DomandaDiRecupero, RispostaDiRecupero)

Acquisto(Animale, Fornitore, DataArrivo, DataAcquisto)

Agriturismo(Codice, Indirizzo)

Animale(Codice, Peso, Sesso, Quarantena, Razza, DataNascita, Altezza, Tipo, Locale, Specie)

AreaTenuta(Codice, Superficie, Agriturismo)

Assegnazione(Prenotazione, Camera, Agriturismo)

Camera(Numero, Agriturismo, Tipo, Costo, DataArrivo)

Cantina(Codice, ClasseEnergetica, Agriturismo)

Caratteristica(Sostanza, CodiceLatte, QuantitaMgMl)

Complicanza(Gestazione, Data, Descrizione)

Attesa(Ordine, Formaggio, Quantita)

CentroSmistamento(Codice, NumeroDipendenti, Latitudine, Longitudine)

Cliente(CodiceDocumento, Nome, Cognome, Registrato, CodiceCarta)

Condizionatore(Codice, Classe, Locale)

Controllo(Codice, Stato, DataProgrammata, DataEffettiva, Esito, Gestazione, Veterinario)

Comporta(Terapia, Controllo)

DispositivoIlluminazione(Codice, GradoProtezione, Lumen, Locale)

Disturbo(Nome, IstanteVisita, Animale, Entita)

Documento(CodiceDocumento, Tipo, Scadenza, EnteRilascio)

Esame(Codice, Nome, Data, Macchinario, Descrizione, CodiceControllo)

Escursione(Codice, OrarioInizio, Costo, Guida)

Esordio(Terapia, Patologia, IstanteVisita, Animale)

Farmaco(Nome, Posologia, PrincipioAttivo)

Fase(Fase, Formaggio, Descrizione)

Finestra(Codice, Orientamento, Locale)

Foraggio(Tipo, PercentualeCereali, PercentualePiante, PercentualeFrutta, ChilocaloriePerChilo, Fibre, Proteine, Glucidi, TipoConservazione)

Formaggio(Tipo, ZonaOrigine, TipoPasta, Deperibilita, Prezzo)

Fornitore(PartitaIVA, RagioneSociale, Nome, Indirizzo)

Gestazione(Codice, DataInterruzione, Complicanze, Responsabile, Istante, Padre, Madre)

Guida(Codice, Nome, Cognome, Agriturismo)
IndicatoreBenessere(Nome)
InterventoPulizia(Istante, Locale, StatoPulizia, LivelloSporcizia, LivelloAzoto, LivelloMetano)
LaboratorioProduzione(Codice, Superficie, NumeroVasche, Agriturismo)
Latte(Codice, Volume, Mungitrice, Animale, Silos)
Lavora(Operaio, Lotto)
Lesione(Tipologia, ParteCorpo, IstanteVisita, Animale, Entita)
Letto(Codice, Capienza, Tipo, Camera, Agriturismo)
Locale(Codice, Pavimentazione, Lunghezza, Larghezza, Altezza, Stalla, Specie, NumeroMassimoAnimali)
Lotto(Codice, Data, Scadenza, DurataProduzione, Silos, Laboratorio)
Magazzino(Codice, Superficie, ClasseEnergetica, Agriturismo)
Mangiatoia(Codice, Larghezza, Lunghezza, Locale)
Monitoraggio(IstanteVisita, Animale, Indicatore, Valore)
Mungitrice(Codice, Latitudine, Longitudine)
Nascita(Animale, Gestazione)
Operaio(Codice, Stipendio, Nome, Cognome)
OrarioSomministrazione(Orario, Farmaco)
Ordine(Codice, Istante, Stato, Cliente)
Pagamento(Codice, Istante, MetodoPagamento, Importo, Prenotazione)
ParametroEffettivo(Nome, Fase, Lotto, Valore)
ParametroProcesso(Nome, Fase, Formaggio, Valore)
Patologia(Nome, ParteCorpo)
Percorso(Hub, Spedizione)
Posizione(Istante, Animale, Latitudine, Longitudine)
Prenota(Prenotazione, Escursione)
Prenotazione(Codice, DataPartenza, DataArrivo, Cliente)
ProdottoCaseario(Codice, Peso, Lotto, Formaggio, Scaffalatura, Ordine, Sconto, FineStagionatura)
ProprietaAcqua(Istante, Abbeveratoio, Sostanza)
Recensione(Codice, Gradimento, Gusto, Qualita, Conservazione, NoteAggiuntive, CodiceOrdine)
Recinzione(Codice, LongitudinelIniziale, LatitudinelIniziale, LongitudineFinale, LatitudineFinale, Tipo, ZonaPascolo, Agriturismo)
Richiede(Prenotazione, Servizio, NumeroGiorni)
Riproduzione(Istante, Padre, Madre, Stato, Supervisore)
Scaffalatura(Codice, Marca, Portata, CapacitaFormaggi, Magazzino, Cantina)
SchedaValutazione(ProdottoCaseario, Odore, Colore, Muffa, ValutazioneGenerale)
ServizioAggiuntivo(Servizio, Costo)
Silos(Codice, LivelloRiempimento, CapacitaEttolitri, Agriturismo)
Somministra(Terapia, Farmaco, GiorniPausa)
SostanzaChimica(Nome, Vitamine, SaliMinerali)
Specie(Nome, Famiglia)
Spedizione(Codice, Stato, DataConsegnaPrevista, DataConsegnaEffettiva, CodiceOrdine)
Stalla(Codice, Zona, Agriturismo)
StatoAbbeveratoio(Istante, Abbeveratoio, QuantitaAcqua)
StatoAria(Istante, Locale, Temperatura, Umidità)

StatoCantina(Istante, Cantina, Ventilazione, Temperatura, Umidita)

StatoMangiatoia(Istante, Mangiatoia, QuantitaCibo, Foraggio)

Step(Numero, Escursione, TempoSosta, AreaTenuta)

Terapia(Codice, DataInizio, Durata, Controllo, IstanteVisita, Animale, Patologia, Gravita, Animale)

Veterinario(Codice, Nome, Cognome, Telefono, Agriturismo)

Visita(Istante, Animale, Veterinario)

ZonaPascolo(Nome, Agriturismo)

6.2) Vincoli di integrità referenziale

- Esiste un vincolo di integrità referenziale tra l'attributo Agriturismo della tabella **Stalla** e l'attributo Codice della tabella **Agriturismo**
- Esiste un vincolo di integrità referenziale tra l'attributo Stalla della tabella **Locale** e l'attributo Codice della tabella **Stalla**
- Esiste un vincolo di integrità referenziale tra l'attributo Specie della tabella **Locale** e l'attributo Nome della tabella **Specie**
- Esiste un vincolo di integrità referenziale tra l'attributo Locale della tabella **InterventoPulizia** e l'attributo Codice della tabella **Locale**
- Esiste un vincolo di integrità referenziale tra l'attributo Locale della tabella **Finestra** e l'attributo Codice della tabella **Locale**
- Esiste un vincolo di integrità referenziale tra l'attributo Locale della tabella **StatoAria** e l'attributo Codice della tabella **Locale**
- Esiste un vincolo di integrità referenziale tra l'attributo Locale della tabella **Mangiatoia** e l'attributo Codice della tabella **Locale**
- Esiste un vincolo di integrità referenziale tra l'attributo Locale della tabella **Abbeveratoio** e l'attributo Codice della tabella **Locale**
- Esiste un vincolo di integrità referenziale tra l'attributo Locale della tabella **Condizionatore** e l'attributo Codice della tabella **Locale**
- Esiste un vincolo di integrità referenziale tra l'attributo Locale della tabella **DispositivoIlluminazione** e l'attributo Codice della tabella **Locale**
- Esiste un vincolo di integrità referenziale tra l'attributo Mangiatoia della tabella **StatoMangiatoia** e l'attributo Codice della tabella **Mangiatoia**
- Esiste un vincolo di integrità referenziale tra l'attributo Foraggio della tabella **StatoMangiatoia** e l'attributo Tipo della tabella **Foraggio**
- Esiste un vincolo di integrità referenziale tra l'attributo Abbeveratoio della tabella **StatoAbbeveratoio** e l'attributo Codice della tabella **Abbeveratoio**
- Esiste un vincolo di integrità referenziale tra l'attributo Istante della tabella **ProprietaAcqua** e l'attributo Istante della tabella **StatoAbbeveratoio**
- Esiste un vincolo di integrità referenziale tra l'attributo Abbeveratoio della tabella **ProprietaAcqua** e l'attributo Abbeveratoio della tabella **StatoAbbeveratoio**
- Esiste un vincolo di integrità referenziale tra l'attributo Sostanza della tabella **ProprietaAcqua** e l'attributo Nome della tabella **SostanzaChimica**
- Esiste un vincolo di integrità referenziale tra l'attributo Locale della tabella **Animale** e l'attributo Codice della tabella **Locale**

- Esiste un vincolo di integrità referenziale tra l'attributo Specie della tabella **Animale** e l'attributo Nome della tabella **Specie**
- Esiste un vincolo di integrità referenziale tra l'attributo Agriturismo della tabella **ZonaPascolo** e l'attributo Codice della tabella **Agriturismo**
- Esiste un vincolo di integrità referenziale tra l'attributo Animale della tabella **Posizione** e l'attributo Codice della tabella **Animale**
- Esiste un vincolo di integrità referenziale tra l'attributo ZonaPascolo della tabella **Recinzione** e l'attributo Nome della tabella **ZonaPascolo**
- Esiste un vincolo di integrità referenziale tra l'attributo Agriturismo della tabella **Recinzione** e l'attributo Agriturismo della tabella **ZonaPascolo**
- Esiste un vincolo di integrità referenziale tra l'attributo Padre della tabella **Riproduzione** e l'attributo Codice della tabella **Animale**
- Esiste un vincolo di integrità referenziale tra l'attributo Madre della tabella **Riproduzione** e l'attributo Codice della tabella **Animale**
- Esiste un vincolo di integrità referenziale tra l'attributo Istante della tabella **Gestazione** e l'attributo Istante della tabella **Riproduzione**
- Esiste un vincolo di integrità referenziale tra l'attributo Padre della tabella **Gestazione** e l'attributo Padre della tabella **Riproduzione**
- Esiste un vincolo di integrità referenziale tra l'attributo Madre della tabella **Gestazione** e l'attributo Madre della tabella **Riproduzione**
- Esiste un vincolo di integrità referenziale tra l'attributo Gestazione della tabella **Controllo** e l'attributo Codice della tabella **Gestazione**
- Esiste un vincolo di integrità referenziale tra l'attributo CodiceControllo della tabella **Esame** e l'attributo Codice della tabella **Controllo**
- Esiste un vincolo di integrità referenziale tra l'attributo Agriturismo della tabella **Veterinario** e l'attributo Codice della tabella **Agriturismo**
- Esiste un vincolo di integrità referenziale tra l'attributo Gestazione della tabella **Animale** e l'attributo Codice della tabella **Gestazione**
- Esiste un vincolo di integrità referenziale tra l'attributo Veterinario della tabella **Controllo** e l'attributo Codice della tabella **Veterinario**
- Esiste un vincolo di integrità referenziale tra l'attributo Responsabile della tabella **Gestazione** e l'attributo Codice della tabella **Veterinario**
- Esiste un vincolo di integrità referenziale tra l'attributo Supervisore della tabella **Riproduzione** e l'attributo Codice della tabella **Veterinario**
- Esiste un vincolo di integrità referenziale tra l'attributo Terapia della tabella **Comporta** e l'attributo Codice della tabella **Terapia**
- Esiste un vincolo di integrità referenziale tra l'attributo Controllo della tabella **Comporta** e l'attributo Codice della tabella **Controllo**
- Esiste un vincolo di integrità referenziale tra l'attributo Animale della tabella **Latte** e l'attributo Codice della tabella **Animale**
- Esiste un vincolo di integrità referenziale tra l'attributo Mungitrice della tabella **Latte** e l'attributo Codice della tabella **Mungitrice**
- Esiste un vincolo di integrità referenziale tra l'attributo CodiceLatte della tabella **Caratteristica** e l'attributo Codice della tabella **Latte**

- Esiste un vincolo di integrità referenziale tra l'attributo Lotto della tabella **ParametroEffettivo** e l'attributo Codice della tabella **Lotto**
- Esiste un vincolo di integrità referenziale tra l'attributo Lotto della tabella **ProdottoCaseario** e l'attributo Codice della tabella **Lotto**
- Esiste un vincolo di integrità referenziale tra l'attributo Formaggio della tabella **Fase** e l'attributo Tipo della tabella **Formaggio**
- Esiste un vincolo di integrità referenziale tra l'attributo Fase della tabella **ParametroProcesso** e l'attributo Fase della tabella **Fase**
- Esiste un vincolo di integrità referenziale tra l'attributo Formaggio della tabella **ParametroProcesso** e l'attributo Formaggio della tabella **Fase**
- Esiste un vincolo di integrità referenziale tra l'attributo Cantina della tabella **StatoCantina** e l'attributo Codice della tabella **Cantina**
- Esiste un vincolo di integrità referenziale tra l'attributo Silos della tabella **Latte** e l'attributo Codice della tabella **Silos**
- Esiste un vincolo di integrità referenziale tra l'attributo Agriturismo della tabella **Silos** e l'attributo Codice della tabella **Agriturismo**
- Esiste un vincolo di integrità referenziale tra l'attributo Agriturismo della tabella **LaboratorioProduzione** e l'attributo Codice della tabella **Agriturismo**
- Esiste un vincolo di integrità referenziale tra l'attributo Silos della tabella **Lotto** e l'attributo Codice della tabella **Silos**
- Esiste un vincolo di integrità referenziale tra l'attributo Laboratorio della tabella **Lotto** e l'attributo Codice della tabella **LaboratorioProduzione**
- Esiste un vincolo di integrità referenziale tra l'attributo Operaio della tabella **Lavora** e l'attributo Codice della tabella **Operaio**
- Esiste un vincolo di integrità referenziale tra l'attributo Lotto della tabella **Lavora** e l'attributo Codice della tabella **Lotto**
- Esiste un vincolo di integrità referenziale tra l'attributo Formaggio della tabella **ProdottoCaseario** e l'attributo Tipo della tabella **Formaggio**
- Esiste un vincolo di integrità referenziale tra l'attributo Scaffalatura della tabella **ProdottoCaseario** e l'attributo Codice della tabella **Scaffalatura**
- Esiste un vincolo di integrità referenziale tra l'attributo Magazzino della tabella **Scaffalatura** e l'attributo Codice della tabella **Magazzino**
- Esiste un vincolo di integrità referenziale tra l'attributo Cantina della tabella **Scaffalatura** e l'attributo Codice della tabella **Cantina**
- Esiste un vincolo di integrità referenziale tra l'attributo Agriturismo della tabella **Magazzino** e l'attributo Codice della tabella **Agriturismo**
- Esiste un vincolo di integrità referenziale tra l'attributo Agriturismo della tabella **Cantina** e l'attributo Codice della tabella **Agriturismo**
- Esiste un vincolo di integrità referenziale tra l'attributo IstanteVisita della tabella **Disturbo** e l'attributo Istante della tabella **Visita**
- Esiste un vincolo di integrità referenziale tra l'attributo Animale della tabella **Disturbo** e l'attributo Animale della tabella **Visita**
- Esiste un vincolo di integrità referenziale tra l'attributo IstanteVisita della tabella **Lesione** e l'attributo Istante della tabella **Visita**

- Esiste un vincolo di integrità referenziale tra l'attributo Animale della tabella **Lesione** e l'attributo Animale della tabella **Visita**
- Esiste un vincolo di integrità referenziale tra l'attributo Farmaco della tabella **OrarioSomministrazione** e l'attributo Nome della tabella **Farmaco**
- Esiste un vincolo di integrità referenziale tra l'attributo Veterinario della tabella **Visita** e l'attributo Codice della tabella **Veterinario**
- Esiste un vincolo di integrità referenziale tra l'attributo Animale della tabella **Visita** e l'attributo Codice della tabella **Animale**
- Esiste un vincolo di integrità referenziale tra l'attributo Animale della tabella **Terapia** e l'attributo Codice della tabella **Animale**
- Esiste un vincolo di integrità referenziale tra l'attributo Terapia della tabella **Esordio** e l'attributo Codice della tabella **Terapia**
- Esiste un vincolo di integrità referenziale tra l'attributo IstanteVisita della tabella **Esordio** e l'attributo Istante della tabella **Visita**
- Esiste un vincolo di integrità referenziale tra l'attributo Animale della tabella **Esordio** e l'attributo Animale della tabella **Visita**
- Esiste un vincolo di integrità referenziale tra l'attributo Patologia della tabella **Esordio** e l'attributo Nome della tabella **Patologia**
- Esiste un vincolo di integrità referenziale tra l'attributo Terapia della tabella **Somministra** e l'attributo Codice della tabella **Terapia**
- Esiste un vincolo di integrità referenziale tra l'attributo Farmaco della tabella **Somministra** e l'attributo Nome della tabella **Farmaco**
- Esiste un vincolo di integrità referenziale tra l'attributo Agriturismo della tabella **Camera** e l'attributo Codice della tabella **Agriturismo**
- Esiste un vincolo di integrità referenziale tra l'attributo Camera della tabella **Letto** e l'attributo Numero della tabella **Camera**
- Esiste un vincolo di integrità referenziale tra l'attributo Agriturismo della tabella **Letto** e l'attributo Agriturismo della tabella **Camera**
- Esiste un vincolo di integrità referenziale tra l'attributo CodiceDocumento della tabella **Account** e l'attributo CodiceDocumento della tabella **Cliente**
- Esiste un vincolo di integrità referenziale tra l'attributo Cliente della tabella **Prenotazione** e l'attributo CodiceDocumento della tabella **Cliente**
- Esiste un vincolo di integrità referenziale tra l'attributo Prenotazione della tabella **Pagamento** e l'attributo Codice della tabella **Prenotazione**
- Esiste un vincolo di integrità referenziale tra l'attributo Guida della tabella **Escursione** e l'attributo Codice della tabella **Guida**
- Esiste un vincolo di integrità referenziale tra l'attributo Agriturismo della tabella **Guida** e l'attributo Codice della tabella **Agriturismo**
- Esiste un vincolo di integrità referenziale tra l'attributo Agriturismo della tabella **AreaTenuta** e l'attributo Codice della tabella **Agriturismo**
- Esiste un vincolo di integrità referenziale tra l'attributo Escursione della tabella **Step** e l'attributo Codice della tabella **Escursione**
- Esiste un vincolo di integrità referenziale tra l'attributo AreaTenuta della tabella **Step** e l'attributo Codice della tabella **AreaTenuta**

- Esiste un vincolo di integrità referenziale tra l'attributo Prenotazione della tabella **Assegnazione** e l'attributo Codice della tabella **Prenotazione**
- Esiste un vincolo di integrità referenziale tra l'attributo Camera della tabella **Assegnazione** e l'attributo Numero della tabella **Camera**
- Esiste un vincolo di integrità referenziale tra l'attributo Agriturismo della tabella **Assegnazione** e l'attributo Agriturismo della tabella **Camera**
- Esiste un vincolo di integrità referenziale tra l'attributo Prenotazione della tabella **Richiede** e l'attributo Codice della tabella **Prenotazione**
- Esiste un vincolo di integrità referenziale tra l'attributo Servizio della tabella **Richiede** e l'attributo Servizio della tabella **ServizioAggiuntivo**
- Esiste un vincolo di integrità referenziale tra l'attributo Prenotazione della tabella **Prenota** e l'attributo Codice della tabella **Prenotazione**
- Esiste un vincolo di integrità referenziale tra l'attributo Escursione della tabella **Prenota** e l'attributo Codice della tabella **Escursione**
- Esiste un vincolo di integrità referenziale tra l'attributo Cliente della tabella **Ordine** e l'attributo CodiceDocumento della tabella **Account**
- Esiste un vincolo di integrità referenziale tra l'attributo Ordine della tabella **Attesa** e l'attributo Codice della tabella **Ordine**
- Esiste un vincolo di integrità referenziale tra l'attributo Formaggio della tabella **Attesa** e l'attributo Tipo della tabella **Formaggio**
- Esiste un vincolo di integrità referenziale tra l'attributo Ordine della tabella **ProdottoCaseario** e l'attributo Codice della tabella **Ordine**
- Esiste un vincolo di integrità referenziale tra l'attributo CodiceOrdine della tabella **Recensione** e l'attributo Codice della tabella **Ordine**
- Esiste un vincolo di integrità referenziale tra l'attributo CodiceOrdine della tabella **Spedizione** e l'attributo Codice della tabella **Ordine**
- Esiste un vincolo di integrità referenziale tra l'attributo Hub della tabella **Percorso** e l'attributo Codice della tabella **CentroSmistamento**
- Esiste un vincolo di integrità referenziale tra l'attributo Spedizione della tabella **Percorso** e l'attributo Codice della tabella **Spedizione**
- Esiste un vincolo di integrità referenziale tra l'attributo IstanteVisita della tabella **Monitoraggio** e l'attributo Istante della tabella **Visita**
- Esiste un vincolo di integrità referenziale tra l'attributo Animale della tabella **Monitoraggio** e l'attributo Animale della tabella **Visita**
- Esiste un vincolo di integrità referenziale tra l'attributo Indicatore della tabella **Monitoraggio** e l'attributo Nome della tabella **IndicatoreBenessere**
- Esiste un vincolo di integrità referenziale tra l'attributo CodiceDocumento della tabella **Documento** e l'attributo CodiceDocumento della tabella **Account**
- Esiste un vincolo di integrità referenziale tra l'attributo Animale della tabella **Acquisto** e l'attributo Codice della tabella **Animale**
- Esiste un vincolo di integrità referenziale tra l'attributo Fornitore della tabella **Acquisto** e l'attributo PartitaIVA della tabella **Fornitore**
- Esiste un vincolo di integrità referenziale tra l'attributo Animale della tabella **Nascita** e l'attributo Codice della tabella **Animale**

- Esiste un vincolo di integrità referenziale tra l'attributo Gestazione della tabella **Nascita** e l'attributo Codice della tabella **Gestazione**
- Esiste un vincolo di integrità referenziale tra l'attributo ProdottoCaseario della tabella **SchedaValutazione** e l'attributo Codice della tabella **ProdottoCaseario**
- Esiste un vincolo di integrità referenziale tra l'attributo Gestazione della tabella **Complicanza** e l'attributo Codice della tabella **Gestazione**

6.3) Vincoli di integrità generici

Analizzando le specifiche di progetto sono stati individuati dei vincoli di integrità generici per vietare determinati scenari incoerenti in cui potrebbero trovarsi i dati.

- (V1) Un locale può ospitare solo animali di una stessa specie
- (V2) Un locale non può ospitare più animali del suo numero massimo
- (V3) Una gestazione può esistere solo se il relativo tentativo di riproduzione ha avuto successo
- (V4) Per acquistare prodotti online un cliente deve fornire il codice fiscale, il numero di telefono e la tipologia, la scadenza e l'ente di rilascio di un documento d'identità (valido)
- (V5) Il documento d'identità inserito non deve essere scaduto
- (V6) Gli ordini devono seguire la sequenza di stati: in processazione, in preparazione, spedito, evaso
- (V7) La data di partenza di una prenotazione non può essere antecedente alla data di arrivo
- (V8) Un animale non può essere nato nell'agriturismo ed essere stato acquistato
- (V9) La data di arrivo di un animale non può essere antecedente alla data di acquisto
- (V10) Una scaffalatura deve trovarsi o in un magazzino o in una cantina
- (V11) Nella coda di attesa saranno presenti soltanto le tipologie e le quantità dei formaggi che mancano per far partire l'ordine
- (V12) La quantità ordinata di un tipo di formaggio deve essere un multiplo di 100 (ovvero il peso della quantità minima di formaggio venduta)
- (V13) La gravità di un esordio ha un valore compreso tra 1 e 10

7. Analisi delle dipendenze funzionali e normalizzazione

Dopo aver tradotto il modello Entità – Relazioni nel modello Relazionale e aver individuato i vincoli di integrità è seguita l'analisi delle dipendenze funzionali e la normalizzazione della base di dati. Come si può vedere già dalla traduzione molte tabelle del database raggiungono già il grado di normalizzazione richiesto dalle specifiche di progetto (Forma Normale di Boyce – Codd).

7.1) Dipendenze funzionali

Account(CodiceDocumento, Username, Password, CodiceFiscale, Indirizzo, Telefono, DataIscrizione, DomandaDiRecupero, RispostaDiRecupero)

CodiceDocumento → Username, Password, CodiceFiscale, Indirizzo, Telefono, DataIscrizione, DomandaDiRecupero, RispostaDiRecupero

Poiché gli attributi a sinistra sono superchiave, **Account** è in BCNF.

Acquisto(Animale, Fornitore, DataArrivo, DataAcquisto)

Animale → Fornitore, DataArrivo, DataAcquisto

Poiché gli attributi a sinistra sono superchiave, **Acquisto** è in BCNF.

Complicanza(Gestazione, Data, Descrizione)

Gestazione → Data, Descrizione

Poiché gli attributi a sinistra sono superchiave, **Complicanza** è in BCNF.

Agriturismo(Codice, Indirizzo)

Codice → Indirizzo

Poiché gli attributi a sinistra sono superchiave, **Agriturismo** è in BCNF.

Animale(Codice, Peso, Sesso, Quarantena, Razza, DataNascita, Altezza, Tipo, Locale, Specie)

Codice → Peso, Sesso, Quarantena, Razza, DataNascita, Altezza, Tipo, Locale, Specie

Razza → Specie

Essendo presente una dipendenza funzionale che può portare a ridondanze, è necessario decomporre la tabella per ottenerne due in **BCNF**:

Animale(Codice, Peso, Sesso, Quarantena, Razza, DataNascita, Altezza, Tipo, Locale)

Razza(Nome, Specie)

La decomposizione non comporta la perdita delle dipendenze funzionali originarie.

Nascita(Animale, Gestazione)

Animale → Gestazione

Poiché gli attributi a sinistra sono superchiave, **Nascita** è in BCNF.

Camera(Numero, Agriturismo, Tipo, Costo, DataArrivo)

Numero, Agriturismo → Tipo, Costo, DataArrivo

Poiché gli attributi a sinistra sono superchiave, **Camera** è in BCNF.

Cliente(CodiceDocumento, Nome, Cognome, Registrato, CodiceCarta)

CodiceDocumento → Nome, Cognome, Registrato, CodiceCarta

Poiché gli attributi a sinistra sono superchiave, **Cliente** è in BCNF.

Documento(CodiceDocumento, Tipo, Scadenza, EnteRilascio)

CodiceDocumento → Tipo, Scadenza, EnteRilascio

Poiché gli attributi a sinistra sono superchiave, **Documento** è in BCNF.

Esordio(Terapia, Patologia, IstanteVisita, Animale)

Terapia → Patologia, IstanteVisita, Animale

Poiché gli attributi a sinistra sono superchiave, **Esordio** è in BCNF.

Stalla(Codice, Zona, Agriturismo)

Codice → Zona, Agriturismo

Poiché gli attributi a sinistra sono superchiave, **Stalla** è in BCNF.

Finestra(Codice, Orientamento, Locale)

Codice → Orientamento, Locale

Poiché gli attributi a sinistra sono superchiave, **Finestra** è in BCNF.

Locale(Codice, Pavimentazione, Lunghezza, Larghezza, Altezza, Stalla, Specie, NumeroMassimoAnimali)

Codice → Pavimentazione, Lunghezza, Larghezza, Altezza, Stalla, Specie, NumeroMassimoAnimali

Poiché gli attributi a sinistra sono superchiave, **Locale** è in BCNF.

InterventoPulizia(Istante, Locale, StatoPulizia, LivelloSporcizia, LivelloAzoto, LivelloMetano)

Istante, Locale → StatoPulizia, LivelloSporcizia, LivelloAzoto, LivelloMetano

Poiché gli attributi a sinistra sono superchiave, **InterventoPulizia** è in BCNF.

StatoAria(Istante, Locale, Temperatura, Umidità)

Istante, Locale → Temperatura, Umidità

Poiché gli attributi a sinistra sono superchiave, **StatoAria** è in BCNF.

Mangiatoia(Codice, Larghezza, Lunghezza, Locale)

Codice → Lunghezza, Larghezza, Locale

Poiché gli attributi a sinistra sono superchiave, **Mangiatoia** è in BCNF.

Abbeveratoio(Codice, Capienza, Locale)

Codice → Capienza, Locale

Poiché gli attributi a sinistra sono superchiave, **Abbeveratoio** è in BCNF.

Condizionatore(Codice, Classe, Locale)

Codice → Classe, Locale

Poiché gli attributi a sinistra sono superchiave, **Condizionatore** è in BCNF.

DispositivoIlluminazione(Codice, GradoProtezione, Lumen, Locale)

Codice → GradoProtezione, Lumen, Locale

Poiché gli attributi a sinistra sono superchiave, **DispositivoIlluminazione** è in BCNF.

StatoMangiatoia(Istante, Mangiatoia, QuantitaCibo, Foraggio)

Istante, Mangiatoia → QuantitaCibo, Foraggio

Poiché gli attributi a sinistra sono superchiave, **StatoMangiatoia** è in BCNF.

Foraggio(Tipo, PercentualeCereali, PercentualePiante, PercentualeFrutta, ChilocaloriePerChilo, Fibre, Proteine, Glucidi, TipoConservazione)

Tipo → PercentualeCereali, PercentualePiante, PercentualeFrutta, ChilocaloriePerChilo, Fibre, Proteine, Glucidi, TipoConservazione

Poiché gli attributi a sinistra sono superchiave, **Foraggio** è in BCNF.

StatoAbbeveratoio(Istante, Abbeveratoio, QuantitaAcqua)

Istante, Abbeveratoio → QuantitaAcqua

Poiché gli attributi a sinistra sono superchiave, **StatoAbbeveratoio** è in BCNF.

SostanzaChimica(Nome, Vitamine, SaliMinerali)

Nome → Vitamine, SaliMinerali

Poiché gli attributi a sinistra sono superchiave, **SostanzaChimica** è in BCNF.

ProprietaAcqua(Istante, Abbeveratoio, Sostanza)

La relazione non presenta dipendenze significative.

Specie(Nome, Famiglia)

Nome → Famiglia

Poiché gli attributi a sinistra sono superchiave, **Specie** è in BCNF.

ZonaPascolo(Nome, Agriturismo)

La relazione non presenta dipendenze significative.

Posizione(Istante, Animale, Latitudine, Longitudine)

Istante, Animale → Latitudine, Longitudine

Poiché gli attributi a sinistra sono superchiave, **Posizione** è in BCNF.

Recinzione(Codice, LongitudinelIniziale, LatitudinelIniziale, LongitudineFinale, LatitudineFinale, Tipo, ZonaPascolo, Agriturismo)

Codice → LongitudineIniziale, LatitudineIniziale, LongitudineFinale, LatitudineFinale, Tipo, ZonaPascolo, Agriturismo
Poiché gli attributi a sinistra sono superchiave, **Recinzione** è in BCNF.

Fornitore(PartitaIVA, RagioneSociale, Nome, Indirizzo)

PartitaIVA → RagioneSociale, Nome, Indirizzo

Poiché gli attributi a sinistra sono superchiave, **Fornitore** è in BCNF.

Riproduzione(Istante, Padre, Madre, Stato, Supervisore)

Istante, Padre, Madre → Stato, Supervisore

Poiché gli attributi a sinistra sono superchiave, **Riproduzione** è in BCNF.

Gestazione(Codice, DataInterruzione, Complicanze, Responsabile, Istante, Padre, Madre)

Codice → DataInterruzione, Complicanze, Responsabile, Istante, Padre, Madre

Poiché gli attributi a sinistra sono superchiave, **Gestazione** è in BCNF.

Controllo(Codice, Stato, DataProgrammata, DataEffettiva, Esito, Gestazione, Veterinario)

Codice → Stato, DataProgrammata, DataEffettiva, Esito, Gestazione, Veterinario

Poiché gli attributi a sinistra sono superchiave, **Controllo** è in BCNF.

Esame(Codice, Nome, Data, Macchinario, Descrizione, CodiceControllo)

Codice → Nome, Data, Macchinario, Descrizione, CodiceControllo

Poiché gli attributi a sinistra sono superchiave, **Esame** è in BCNF.

Veterinario(Codice, Nome, Cognome, Telefono, Agriturismo)

Codice → Nome, Cognome, Telefono, Agriturismo

Poiché gli attributi a sinistra sono superchiave, **Veterinario** è in BCNF.

Mungitrice(Codice, Latitudine, Longitudine)

Codice → Latitudine, Longitudine

Poiché gli attributi a sinistra sono superchiave, **Mungitrice** è in BCNF.

Latte(Codice, Volume, Mungitrice, Animale, Silos)

Codice → Volume, Mungitrice, Animale, Silos

Poiché gli attributi a sinistra sono superchiave, **Latte** è in BCNF.

Caratteristica(Sostanza, CodiceLatte, QuantitaMgMI)

Sostanza, CodiceLatte → QuantitaMgMI

Poiché gli attributi a sinistra sono superchiave, **Caratteristica** è in BCNF.

Silos(Codice, LivelloRiempimento, CapacitaEttolitri, Agriturismo)

Codice → LivelloRiempimento, CapacitaEttolitri, Agriturismo

Poiché gli attributi a sinistra sono superchiave, **Silos** è in BCNF.

LaboratorioProduzione(Codice, Superficie, NumeroVasche, Agriturismo)

Codice → Superficie, NumeroVasche, Agriturismo

Poiché gli attributi a sinistra sono superchiave, **LaboratorioProduzione** è in BCNF.

Lotto(Codice, Data, Scadenza, DurataProduzione, Silos, Laboratorio)

Codice → Data, Scadenza, DurataProduzione, Silos, Laboratorio

Poiché gli attributi a sinistra sono superchiave, **Lotto** è in BCNF.

ParametroEffettivo(Nome, Fase, Lotto, Valore)

Nome, Fase, Lotto → Valore

Poiché gli attributi a sinistra sono superchiave, **ParametroEffettivo** è in BCNF.

Operaio(Codice, Stipendio, Nome, Cognome)

Codice → Stipendio, Nome, Cognome

Poiché gli attributi a sinistra sono superchiave, **Operaio** è in BCNF.

ProdottoCaseario(Codice, Peso, Lotto, Formaggio, Scaffalatura, Ordine, Reso, Sconto, FineStagionatura)

Codice → Peso, Lotto, Formaggio, Scaffalatura, Ordine, Reso, Sconto, FineStagionatura

Poiché gli attributi a sinistra sono superchiave, **ProdottoCaseario** è in BCNF.

Formaggio(Tipo, ZonaOrigine, TipoPasta, Deperibilita, Prezzo)

Tipo → ZonaOrigine, TipoPasta, Deperibilita, Prezzo

Poiché gli attributi a sinistra sono superchiave, **Formaggio** è in BCNF.

Fase(Fase, Formaggio, Descrizione)

Fase, Formaggio → Descrizione

Poiché gli attributi a sinistra sono superchiave, **Fase** è in BCNF.

ParametroProcesso(Nome, Fase, Formaggio, Valore)

Nome, Fase, Formaggio → Valore

Poiché gli attributi a sinistra sono superchiave, **ParametroProcesso** è in BCNF.

Magazzino(Codice, Superficie, ClasseEnergetica, Agriturismo)

Codice → Superficie, ClasseEnergetica, Agriturismo

Poiché gli attributi a sinistra sono superchiave, **Magazzino** è in BCNF.

Cantina(Codice, ClasseEnergetica, Agriturismo)

Codice → ClasseEnergetica, Agriturismo

Poiché gli attributi a sinistra sono superchiave, **Cantina** è in BCNF.

Comporta(Terapia, Controllo)

Terapia → Controllo

Poiché gli attributi a sinistra sono superchiave, **Comporta** è in BCNF.

Scaffalatura(Codice, Marca, Portata, CapacitaFormaggi, Magazzino, Cantina)
Codice → Marca, Portata, CapacitaFormaggi, Magazzino, Cantina
Poiché gli attributi a sinistra sono superchiave, **Scaffalatura** è in BCNF.

StatoCantina(Istante, Cantina, Ventilazione, Temperatura, Umidita)
Istante, Cantina → Ventilazione, Temperatura, Umidita
Poiché gli attributi a sinistra sono superchiave, **StatoCantina** è in BCNF.

Lavora(Operaio, Lotto)
La relazione non presenta dipendenze significative.

Visita(Istante, Animale, Veterinario)
Istante, Animale → Veterinario
Poiché gli attributi a sinistra sono superchiave, **Visita** è in BCNF.

Disturbo(Nome, IstanteVisita, Animale, Entita)
Nome, IstanteVisita, Animale → Entita
Poiché gli attributi a sinistra sono superchiave, **Disturbo** è in BCNF.

Lesione(Tipologia, ParteCorpo, IstanteVisita, Animale, Entita)
Tipologia, ParteCorpo, IstanteVisita, Animale → Entita
Poiché gli attributi a sinistra sono superchiave, **Lesione** è in BCNF.

Patologia(Nome, ParteCorpo)
Nome → ParteCorpo
Poiché gli attributi a sinistra sono superchiave, **Patologia** è in BCNF.

Terapia(Codice, DataInizio, Durata, IstanteVisita, Animale, Patologia, Gravita)
Codice → DataInizio, Durata, IstanteVisita, Animale, Patologia, Gravita
Poiché gli attributi a sinistra sono superchiave, **Terapia** è in BCNF.

Farmaco(Nome, Posologia, PrincipioAttivo)
Nome → Posologia, PrincipioAttivo
Poiché gli attributi a sinistra sono superchiave, **Farmaco** è in BCNF.

OrarioSomministrazione(Orario, Farmaco)
La relazione non presenta dipendenze significative.

Somministra(Terapia, Farmaco, GiorniPausa)
Terapia, Farmaco → GiorniPausa
Poiché gli attributi a sinistra sono superchiave, **Somministra** è in BCNF.

Letto(Codice, Capienza, Tipo, Camera, Agriturismo)

Codice → Capienza, Tipo, Numero, Agriturismo

Tipo → Capienza

Essendo presente una dipendenza funzionale che può portare a ridondanze, è necessario decomporre la tabella per ottenerne due in **BCNF**:

Letto(Codice, Tipo, Camera, Agriturismo)

Capienza(Tipo, Capienza)

La decomposizione non comporta la perdita delle dipendenze funzionali originarie.

Prenotazione(Codice, DataPartenza, DataArrivo, Cliente)

Codice → DataPartenza, DataArrivo, Cliente

Poiché gli attributi a sinistra sono superchiave, **Prenotazione** è in BCNF.

ServizioAggiuntivo(Servizio, Costo)

Servizio → Costo

Poiché gli attributi a sinistra sono superchiave, **ServizioAggiuntivo** è in BCNF.

Pagamento(Codice, Istante, MetodoPagamento, Importo, Prenotazione)

Codice → Istante, MetodoPagamento, Importo, Prenotazione

Poiché gli attributi a sinistra sono superchiave, **Pagamento** è in BCNF.

Escursione(Codice, OrarioInizio, Costo, Guida)

Codice → OrarioInizio, Costo, Guida

Poiché gli attributi a sinistra sono superchiave, **Escursione** è in BCNF.

Guida(Codice, Nome, Cognome, Agriturismo)

Codice → Nome, Cognome, Agriturismo

Poiché gli attributi a sinistra sono superchiave, **Guida** è in BCNF.

Step(Numero, Escursione, TempoSosta, AreaTenuta)

Numero, Escursione → TempoSosta, AreaTenuta

Poiché gli attributi a sinistra sono superchiave, **Step** è in BCNF.

AreaTenuta(Codice, Superficie, Agriturismo)

Codice → Superficie, Agriturismo

Poiché gli attributi a sinistra sono superchiave, **AreaTenuta** è in BCNF.

Assegnazione(Prenotazione, Camera, Agriturismo)

La relazione non presenta dipendenze significative.

Richiede(Prenotazione, Servizio, NumeroGiorni)

Prenotazione, Servizio → NumeroGiorni

Poiché gli attributi a sinistra sono superchiave, **Richiede** è in BCNF.

Prenota(Prenotazione, Escursione)

La relazione non presenta dipendenze significative.

Ordine(Codice, Istante, Stato, Cliente)

Codice → Istante, Stato, Cliente

Poiché gli attributi a sinistra sono superchiave, **Ordine** è in BCNF.

Recensione(Codice, Gradimento, Gusto, Qualita, Conservazione, NoteAggiuntive, CodiceOrdine)

Codice → Gradimento, Gusto, Qualita, Conservazione, NoteAggiuntive, CodiceOrdine

Poiché gli attributi a sinistra sono superchiave, **Recensione** è in BCNF.

Spedizione(Codice, Stato, DataConsegnaPrevista, DataConsegnaEffettiva, CodiceOrdine)

Codice → Stato, DataConsegnaPrevista, DataConsegnaEffettiva, CodiceOrdine

Poiché gli attributi a sinistra sono superchiave, **Spedizione** è in BCNF.

CentroSmistamento(Codice, NumeroDipendenti, Latitudine, Longitudine)

Codice → NumeroDipendenti, Latitudine, Longitudine

Poiché gli attributi a sinistra sono superchiave, **CentroSmistamento** è in BCNF.

Attesa(Ordine, Formaggio, Quantita)

Ordine, Formaggio → Quantita

Poiché gli attributi a sinistra sono superchiave, **Attesa** è in BCNF.

SchedaValutazione(ProdottoCaseario, Odore, Colore, Muffa, ValutazioneGenerale)

Ordine, Formaggio → Quantita

Poiché gli attributi a sinistra sono superchiave, **Attesa** è in BCNF.

Percorso(Hub, Spedizione)

La relazione non presenta dipendenze significative.

IndicatoreBenessere(Nome)

La relazione non presenta dipendenze significative.

Monitoraggio(IstanteVisita, Animale, Indicatore, Valore)

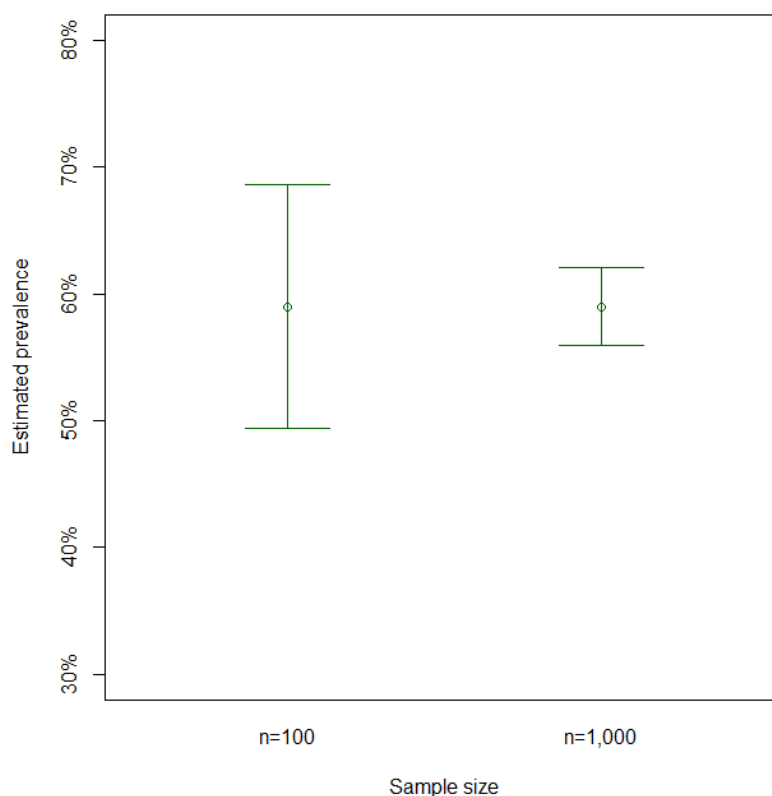
IstanteVisita, Animale, Indicatore → Valore

Poiché gli attributi a sinistra sono superchiave, **Monitoraggio** è in BCNF.

8. Analytics

8.1) Comportamento degli animali

È stata eseguita un'analisi approfondita sul comportamento degli animali per poterne dedurre lo stato di benessere, possibili *mancanze*, sintomatologie che indicassero sofferenze/patologie e la mancata integrazione con gli altri animali.



8.1.1) Spostamento medio

È stato misurato lo **scostamento** tra il numero di **spostamenti** di ciascun animale durante le attività di pascolo e la media degli spostamenti di tutti gli animali facenti parte dello stesso agriturismo. Questo valore ci permette di capire la differenza quantitativa di movimento (in eccesso o in difetto) di un animale rispetto a tutti gli altri.

La media su cui calcolare lo scostamento non è stata valutata sulle singole zone di pascolo per ragioni statistiche (1) né sulla totalità degli agriturismi per ragioni di spazi (2). Inoltre, come intervallo temporale, è stata considerata **una settimana**: lo scostamento non dovrebbe essere influenzato dai malesseri troppo datati non più presenti né da una passeggera (1/2 giorni) sofferenza dell'animale.

- 1) È stato calcolato che se la media venisse misurata sulle singole aree di pascolo il numero di posizioni su cui basarsi sarebbe 1/15 delle posizioni che avremmo se invece considerassimo agriturismo per agriturismo. Ciò comporterebbe una notevole riduzione della precisione della stima come è possibile notare dal grafico: maggiore è il numero di campioni considerati, più piccolo sarà il margine di errore (e quindi maggiori saranno le probabilità che il valore medio ottenuto sia quello reale).

- 2) Non sono state aggregate le posizioni di tutti agriturismi poiché se un agriturismo fosse notevolmente più piccolo di un altro (con conseguenti aree di pascolo più piccole) sarebbe normale che il numero di spostamenti degli animali di quell'agriturismo siano inferiori agli spostamenti degli animali che hanno la possibilità di pascolare in zone molto più ampie.

Le informazioni sullo scostamento ci danno informazioni sullo stato di salute dell'animale: potrebbe essere troppo anziano per pascolare e non riuscire quasi più a muoversi, potrebbe indicare un disturbo comportamentale che lo fa muovere in eccesso o una lesione che, al contrario, non gli permette di muoversi liberamente.

Allo stesso modo potrebbero anche darci informazioni sull'*integrazione* di un animale appena acquistato con gli altri simili. Sicuramente nel caso si muova poco o resti addirittura fermo potrebbe essere un segno di mancata *interazione con l'ambiente* che lo circonda.

8.1.2) Distanza dagli altri animali

Misurando la **distanza** tra gli **animali** di uno stesso locale (che pascoleranno in una stessa zona) sarà possibile:

1. Capire se due animali si stiano **corteggiando** (in questo caso verranno presi in considerazione soltanto gli animali di sesso opposto);
2. Individuare casi di **isolamento** degli animali causati da svariati fattori che riguardano sempre la salute dell'animale. Una di queste potrebbe essere un farmaco non *tollerato* dall'animale che ne ha portato ad una compromissione dello stato fisico, una gravidanza inaspettata e non monitorata che sta causando problemi alla gestante ecc.

Per il calcolo della distanza vengono presi in considerazione i **sette giorni** precedenti in quanto non ci interessa avere delle informazioni troppo datate che potrebbero *compromettere* quelle attuali sicuramente più realistiche.

Il report sui corteggiamenti è fondamentale per poter capire se due animali siano in sintonia e siano quindi più apprezzabili per un tentativo di riproduzione, se venissero ritenuti idonei dopo svariate valutazioni. Allo stesso modo può anche mettere in evidenza se due animali stanno sempre distanti fra loro che potrebbe portare a pensare a dei dissidi di vario genere.

Per quanto riguarda l'individuazione dei casi di isolamento invece viene calcolata la distanza media di un animale da tutti gli altri del suo stesso locale per ciascuno dei sette giorni. Vengono poi conteggiati i giorni in cui questa distanza è superiore o uguale a 6 metri (distanza di isolamento). In base ai giorni ottenuti la direzione potrà prendere varie decisioni come far visitare l'animale, variarne il nutrimento con foraggi specifici, sali minerali e vitamine, tentare di spostarlo in un altro locale e se non niente dovesse portare ad un risultato potrebbe anche decidere di venderlo.

8.2) Materialized view

Nel database sono state inserite le seguenti materialized view per poter avere una visione migliore e più chiara di alcuni insiemi di query, principalmente nei report.

MV_ATTIVITA (Animale, NumeroSpostamenti, SpostamentoPerc, TipoScostamento)

MV_CORTEGGIAMENTO (IstanteRilevazione1, Animale1, IstanteRilevazione2, Animale2, Sesso1, Sesso2, Distanza)

MV_ISOLAMENTO (Animale, Tipo, GiorniIsolamento)

9. Implementazione su DBMS

L'ultimo passaggio del progetto è stato l'implementazione della base di dati su DBMS di tutte le tabelle, le operazioni e i vincoli.

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

-- -----
-- Creazione tabella Agriturismo
-- -----
DROP TABLE IF EXISTS `Agriturismo`;
CREATE TABLE `Agriturismo` (
  `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
  `Indirizzo` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`Codice`)
);

-- -----
-- Creazione tabella Stalla
-- -----
DROP TABLE IF EXISTS `Stalla`;
CREATE TABLE `Stalla` (
  `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
  `Zona` VARCHAR(30) NOT NULL,
  `Agriturismo` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`Codice`)
);

-- -----
-- Creazione tabella Locale
-- -----
DROP TABLE IF EXISTS `Locale`;
CREATE TABLE `Locale` (
  `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
  `Pavimentazione` VARCHAR(30) NOT NULL,
  `Lunghezza` SMALLINT UNSIGNED NOT NULL,
```

```

        `Larghezza` SMALLINT UNSIGNED NOT NULL,
        `Altezza` FLOAT UNSIGNED NOT NULL,
        `Stalla` INT UNSIGNED NOT NULL,
        `Specie` VARCHAR(50) NOT NULL,
        `NumeroMassimoAnimali` SMALLINT UNSIGNED NOT NULL,
        PRIMARY KEY (`Codice`)
    );

-- -----
-- Creazione tabella InterventoPulizia
-- -----
DROP TABLE IF EXISTS `InterventoPulizia`;
CREATE TABLE `InterventoPulizia` (
    `Istante` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    `Locale` INT UNSIGNED NOT NULL,
    `StatoPulizia` VARCHAR(10) NOT NULL, -- Richiesto o Effettuato
    `LivelloSporcizia` TINYINT UNSIGNED NOT NULL, -- Da 1 a 100
    `LivelloAzoto` TINYINT UNSIGNED NOT NULL, -- Da 1 a 100
    `LivelloMetano` TINYINT UNSIGNED NOT NULL, -- Da 1 a 100
    PRIMARY KEY (`Istante`, `Locale`)
);

-- -----
-- Creazione tabella Finestra
-- -----
DROP TABLE IF EXISTS `Finestra`;
CREATE TABLE `Finestra` (
    `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
    `Orientamento` VARCHAR(5) NOT NULL, -- Nord, Sud, Est, Ovest
    `Locale` INT UNSIGNED NOT NULL,
    PRIMARY KEY (`Codice`)
);

-- -----
-- Creazione tabella StatoAria
-- -----
DROP TABLE IF EXISTS `StatoAria`;
CREATE TABLE `StatoAria` (
    `Istante` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    `Locale` INT UNSIGNED NOT NULL,
    `Temperatura` TINYINT UNSIGNED NOT NULL, -- In gradi Celsius
    `Umidita` TINYINT UNSIGNED NOT NULL, -- In % (da 0 a 100)
    PRIMARY KEY (`Istante`, `Locale`)
);

-- -----
-- Creazione tabella Mangiatoia
-- -----
DROP TABLE IF EXISTS `Mangiatoia`;
CREATE TABLE `Mangiatoia` (
    `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
    `Lunghezza` TINYINT UNSIGNED NOT NULL, -- In metri
    `Larghezza` TINYINT UNSIGNED NOT NULL, -- In metri
    `Locale` INT UNSIGNED NOT NULL,
    PRIMARY KEY (`Codice`)
);

-- -----
-- Creazione tabella Abbeveratoio
-- -----

```

```

DROP TABLE IF EXISTS `Abbeveratoio`;
CREATE TABLE `Abbeveratoio` (
  `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
  `Capienza` SMALLINT UNSIGNED NOT NULL, -- Litri d'acqua che può
  contenere
  `Locale` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`Codice`)
);

-- -----
-- Creazione tabella Condizionatore
-- -----
DROP TABLE IF EXISTS `Condizionatore`;
CREATE TABLE `Condizionatore` (
  `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
  `Classe` VARCHAR(10) NOT NULL,
  `Locale` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`Codice`)
);

-- -----
-- Creazione tabella DispositivoIlluminazione
-- -----
DROP TABLE IF EXISTS `DispositivoIlluminazione`;
CREATE TABLE `DispositivoIlluminazione` (
  `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
  `GradoProtezione` VARCHAR(4) NOT NULL,
  `Lumen` SMALLINT UNSIGNED NOT NULL,
  `Locale` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`Codice`)
);

-- -----
-- Creazione tabella Foraggio
-- -----
DROP TABLE IF EXISTS `Foraggio`;
CREATE TABLE `Foraggio` (
  `Tipo` VARCHAR(50) NOT NULL,
  `PercentualeCereali` TINYINT UNSIGNED NOT NULL,
  `PercentualePiante` TINYINT UNSIGNED NOT NULL,
  `PercentualeFrutta` TINYINT UNSIGNED NOT NULL,
  `Chilocalorie` SMALLINT UNSIGNED NOT NULL, -- Kcal al kg
  `Fibre` SMALLINT UNSIGNED NOT NULL, -- Grammi al kg
  `Proteine` SMALLINT UNSIGNED NOT NULL, -- Grammi al kg
  `Glucidi` SMALLINT UNSIGNED NOT NULL, -- Grammi al kg
  `TipoConservazione` VARCHAR(10) NOT NULL,
  PRIMARY KEY (`Tipo`)
);

-- -----
-- Creazione tabella StatoMangiatoia
-- -----
DROP TABLE IF EXISTS `StatoMangiatoia`;
CREATE TABLE `StatoMangiatoia` (
  `Istante` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `Mangiatoia` INT UNSIGNED NOT NULL,
  `QuantitaCibo` SMALLINT UNSIGNED NOT NULL, -- In kg
  `Foraggio` VARCHAR(50) NULL,
  PRIMARY KEY (`Istante`, `Mangiatoia`)
);

```

```

-- -----
-- Creazione tabella StatoAbbeveratoio
-- -----
DROP TABLE IF EXISTS `StatoAbbeveratoio`;
CREATE TABLE `StatoAbbeveratoio` (
  `Istante` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `Abbeveratoio` INT UNSIGNED NOT NULL,
  `QuantitaAcqua` SMALLINT UNSIGNED NOT NULL,
  PRIMARY KEY (`Istante`, `Abbeveratoio`)
);

-- -----
-- Creazione tabella SostanzaChimica
-- -----
DROP TABLE IF EXISTS `SostanzaChimica`;
CREATE TABLE `SostanzaChimica` (
  `Nome` VARCHAR(50) NOT NULL,
  `Vitamine` TINYINT UNSIGNED NOT NULL, -- Mg/l
  `SaliMinerali` TINYINT UNSIGNED NOT NULL, -- Mg/l
  PRIMARY KEY (`Nome`)
);

-- -----
-- Creazione tabella ProprietaAcqua
-- -----
DROP TABLE IF EXISTS `ProprietaAcqua`;
CREATE TABLE `ProprietaAcqua` (
  `Istante` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `Abbeveratoio` INT UNSIGNED NOT NULL,
  `Sostanza` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`Istante`, `Abbeveratoio`, `Sostanza`)
);

-- -----
-- Creazione tabella Specie
-- -----
DROP TABLE IF EXISTS `Specie`;
CREATE TABLE `Specie` (
  `Nome` VARCHAR(50) NOT NULL,
  `Famiglia` VARCHAR(30) NOT NULL,
  PRIMARY KEY (`Nome`)
);

-- -----
-- Creazione tabella Fornitore
-- -----
DROP TABLE IF EXISTS `Fornitore`;
CREATE TABLE `Fornitore` (
  `PartitaIVA` VARCHAR(11) NOT NULL,
  `RagioneSociale` VARCHAR(50) NOT NULL,
  `Nome` VARCHAR(50) NOT NULL,
  `Indirizzo` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`PartitaIVA`)
);

-- -----
-- Creazione tabella Acquisto
-- -----
DROP TABLE IF EXISTS `Acquisto`;

```

```

CREATE TABLE `Acquisto` (
    `Animale` INT UNSIGNED NOT NULL,
    `Fornitore` VARCHAR(50) NOT NULL,
    `DataArrivo` DATE NOT NULL,
    `DataAcquisto` DATE NOT NULL,
    PRIMARY KEY (`Animale`)
);

-- Controllo che un animale sia nato in agriturismo oppure acquistato
(Vincolo generico n.8)
DELIMITER $$
DROP TRIGGER IF EXISTS `ControlloAcquisto` $$
CREATE TRIGGER `ControlloAcquisto` BEFORE INSERT ON `Acquisto` FOR EACH ROW
BEGIN
    -- Controllo se è acquistato o nato
    IF (SELECT Tipo FROM Animale WHERE Codice = NEW.Animale) = 'Nato' THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Il tipo dell''animale è "nato", quindi
non può essere stato acquistato!';
    END IF;
END $$

-- Controllo anche l'aggiornamento
DROP TRIGGER IF EXISTS `ControlloAcquisto_U` $$
CREATE TRIGGER `ControlloAcquisto_U` BEFORE UPDATE ON `Acquisto` FOR EACH
ROW
BEGIN
    -- Controllo se è acquistato o nato
    IF (SELECT Tipo FROM Animale WHERE Codice = NEW.Animale) = 'Nato' THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Il tipo dell''animale è "nato", quindi
non può essere stato acquistato!';
    END IF;
END $$
DELIMITER ;

-- Controllo che la data di arrivo non sia precedente alla data d'acquisto
(Vincolo generico n.9)
DELIMITER $$
DROP TRIGGER IF EXISTS `ControlloArrivo` $$
CREATE TRIGGER `ControlloArrivo` BEFORE INSERT ON `Acquisto` FOR EACH ROW
BEGIN
    IF NEW.DataAcquisto > NEW.DataArrivo THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'La data di arrivo non può essere antecedente
a quella di acquisto!';
    END IF;
END $$

-- Controllo anche l'aggiornamento
DROP TRIGGER IF EXISTS `ControlloArrivo_U` $$
CREATE TRIGGER `ControlloArrivo_U` BEFORE UPDATE ON `Acquisto` FOR EACH ROW
BEGIN
    IF NEW.DataAcquisto > NEW.DataArrivo THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'La data di arrivo non può essere antecedente
a quella di acquisto!';
    END IF;
END $$
DELIMITER ;

```

```

-- -----
-- Creazione tabella Nascita
-- -----
DROP TABLE IF EXISTS `Nascita`;
CREATE TABLE `Nascita` (
    `Animale` INT UNSIGNED NOT NULL,
    `Gestazione` INT UNSIGNED NOT NULL,
    PRIMARY KEY (`Animale`)
);

-- Controllo che un animale sia nato in agriturismo oppure acquistato
(Vincolo generico n.8)
DELIMITER $$
DROP TRIGGER IF EXISTS `ControlloNascita` $$
CREATE TRIGGER `ControlloNascita` BEFORE INSERT ON `Nascita` FOR EACH ROW
BEGIN
    -- Controllo se è acquistato o nato
    IF (SELECT Tipo FROM Animale WHERE Codice = NEW.Animale) = 'Acquistato'
    THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Il tipo dell''animale è "acquistato",
quindi non può essere nato nell''agriturismo!';
        END IF;
    END $$

-- Controllo anche l'aggiornamento
DROP TRIGGER IF EXISTS `ControlloNascita_U` $$
CREATE TRIGGER `ControlloNascita_U` BEFORE UPDATE ON `Nascita` FOR EACH ROW
BEGIN
    -- Controllo se è acquistato o nato
    IF (SELECT Tipo FROM Animale WHERE Codice = NEW.Animale) = 'Acquistato'
    THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Il tipo dell''animale è "acquistato",
quindi non può essere nato nell''agriturismo!';
        END IF;
    END $$
DELIMITER ;

-- -----
-- Creazione tabella Animale
-- -----
DROP TABLE IF EXISTS `Animale`;
CREATE TABLE `Animale` (
    `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
    `Peso` SMALLINT UNSIGNED NOT NULL, -- In kg
    `Sesso` VARCHAR(1) NOT NULL, -- M o S
    `Quarantena` TINYINT(1) NOT NULL DEFAULT FALSE,
    `Razza` VARCHAR(30) NOT NULL,
    `DataNascita` DATE NOT NULL,
    `Altezza` TINYINT UNSIGNED NOT NULL,
    `Tipo` VARCHAR(10) NOT NULL, -- Acquistato o Nato
    `Locale` INT UNSIGNED NOT NULL,
    PRIMARY KEY (`Codice`)
);

-- Un locale può ospitare solo animali di una stessa specie (Vincolo
generico n.1)
DELIMITER $$

```

```

DROP TRIGGER IF EXISTS `ControlloSpecie` $$
CREATE TRIGGER `ControlloSpecie` BEFORE INSERT ON `Animale` FOR EACH ROW
BEGIN
    DECLARE specie VARCHAR(50);

    SELECT R.Specie INTO specie
    FROM Razza R
    WHERE R.Nome = NEW.Razza;

    -- Se la specie che ospita il locale è diversa
    IF (SELECT L.Specie FROM Locale L WHERE L.Codice = NEW.Locale) <>
specie THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Un locale può ospitare solo animali della
stessa specie!';
    END IF;
END $$

-- Controlla anche quando gli animali vengono cambiati di locale
DROP TRIGGER IF EXISTS `ControlloSpecie_U` $$
CREATE TRIGGER `ControlloSpecie_U` BEFORE UPDATE ON `Animale` FOR EACH ROW
BEGIN
    DECLARE specie VARCHAR(50);

    -- Se è stato modificato il locale
    IF OLD.Locale <> NEW.Locale THEN

        SELECT R.Specie INTO specie
        FROM Razza R
        WHERE R.Nome = NEW.Razza;

        -- Se la specie che ospita il locale è diversa
        IF (SELECT L.Specie FROM Locale L WHERE L.Codice = NEW.Locale) <>
specie THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Un locale può ospitare solo animali
della stessa specie!';
        END IF;
    END IF;
END $$
DELIMITER ;

-- Un locale può ospitare un numero massimo di animali (Vincolo generico
n.2)
DELIMITER $$
DROP TRIGGER IF EXISTS `ControlloSpazio` $$
CREATE TRIGGER `ControlloSpazio` BEFORE INSERT ON `Animale` FOR EACH ROW
BEGIN
    -- Numero degli animali presenti nel locale in questione
    DECLARE numero_animali SMALLINT UNSIGNED DEFAULT 0;
    -- Numero massimo di animali ospitabili
    DECLARE numero_massimo SMALLINT UNSIGNED DEFAULT 0;

    SELECT COUNT(*) INTO numero_animali
    FROM Animale A
    WHERE A.Locale = NEW.Locale;

    SELECT L.NumeroMassimoAnimali INTO numero_massimo
    FROM Locale L
    WHERE L.Codice = NEW.Locale;

```



```

-- Limite superato
IF numero_animali >= numero_massimo THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Il locale è già pieno!';
END IF;
END $$

-- Controlla anche quando gli animali vengono cambiati di locale
DROP TRIGGER IF EXISTS `ControlloSpazio_U` $$
CREATE TRIGGER `ControlloSpazio_U` BEFORE UPDATE ON `Animale` FOR EACH ROW
BEGIN
    -- Numero degli animali presenti nel locale in questione
    DECLARE numero_animali SMALLINT UNSIGNED DEFAULT 0;
    -- Numero massimo di animali ospitabili
    DECLARE numero_massimo SMALLINT UNSIGNED DEFAULT 0;

    -- Se è stato modificato il locale
    IF OLD.Locale <> NEW.Locale THEN

        -- Numero animali attualmente presenti nel locale
        SELECT COUNT(*) INTO numero_animali
        FROM Animale A
        WHERE A.Locale = NEW.Locale;

        SELECT L.NumeroMassimoAnimali INTO numero_massimo
        FROM Locale L
        WHERE L.Codice = NEW.Locale;

        -- Limite superato
        IF numero_animali >= numero_massimo THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Il nuovo locale è già pieno!';
        END IF;
    END IF;
END $$
DELIMITER ;

-- -----
-- Creazione tabella Razza
-- -----
DROP TABLE IF EXISTS `Razza`;
CREATE TABLE `Razza` (
    `Nome` VARCHAR(50) NOT NULL,
    `Specie` VARCHAR(50) NOT NULL,
    PRIMARY KEY (`Nome`)
);

-- -----
-- Creazione tabella ZonaPascolo
-- -----
DROP TABLE IF EXISTS `ZonaPascolo`;
CREATE TABLE `ZonaPascolo` (
    `Nome` VARCHAR(50) NOT NULL,
    `Agriturismo` INT UNSIGNED NOT NULL,
    PRIMARY KEY (`Nome`, `Agriturismo`)
);

-- -----
-- Creazione tabella Posizione

```

```

-- -----
DROP TABLE IF EXISTS `Posizione`;
CREATE TABLE `Posizione` (
  `Istante` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `Animale` INT UNSIGNED NOT NULL,
  `Latitudine` DECIMAL(7, 5) NOT NULL, -- 7 = cifre totali massime, 5 =
cifre dopo la virgola che ci danno una precisione di 1 metro
  `Longitudine` DECIMAL(8, 5) NOT NULL, -- 8 = cifre totali massime, 5 =
cifre dopo la virgola che ci danno una precisione di 1 metro
  PRIMARY KEY (`Istante`, `Animale`)
);

-- -----
-- Creazione tabella Recinzione
-- -----
DROP TABLE IF EXISTS `Recinzione`;
CREATE TABLE `Recinzione` (
  `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
  `LatitudineIniziale` DECIMAL(7, 5) NOT NULL, -- 7 = cifre totali
massime, 5 = cifre dopo la virgola che ci danno una precisione di 1 metro
  `LongitudineIniziale` DECIMAL(8, 5) NOT NULL, -- 8 = cifre totali
massime, 5 = cifre dopo la virgola che ci danno una precisione di 1 metro
  `LatitudineFinale` DECIMAL(7, 5) NOT NULL, -- 7 = cifre totali massime,
5 = cifre dopo la virgola che ci danno una precisione di 1 metro
  `LongitudineFinale` DECIMAL(8, 5) NOT NULL, -- 8 = cifre totali massime,
5 = cifre dopo la virgola che ci danno una precisione di 1 metro
  `Tipo` VARCHAR(10) NOT NULL, -- Fissa (recinzione esterna) o Mobile
(recinzioni interne alla fissa che possono essere spostate)
  `ZonaPascolo` VARCHAR(50) NOT NULL,
  `Agriturismo` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`Codice`)
);

-- -----
-- Creazione tabella Riproduzione
-- -----
DROP TABLE IF EXISTS `Riproduzione`;
CREATE TABLE `Riproduzione` (
  `Istante` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `Padre` INT UNSIGNED NOT NULL,
  `Madre` INT UNSIGNED NOT NULL,
  `Stato` VARCHAR(10) NOT NULL, -- Successo, Insuccesso o In attesa
  `Supervisore` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`Istante`, `Padre`, `Madre`)
);

-- -----
-- Creazione tabella Complicanza
-- -----
DROP TABLE IF EXISTS `Complicanza`;
CREATE TABLE `Complicanza` (
  `Gestazione` INT UNSIGNED NOT NULL,
  `Data` DATE NOT NULL,
  `Descrizione` VARCHAR(1000) NOT NULL,
  PRIMARY KEY (`Gestazione`)
);

-- -----
-- Creazione tabella Gestazione
-- -----

```

```

DROP TABLE IF EXISTS `Gestazione`;
CREATE TABLE `Gestazione` (
  `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
  `Responsabile` INT UNSIGNED NOT NULL,
  `Istante` TIMESTAMP NOT NULL,
  `Padre` INT UNSIGNED NOT NULL,
  `Madre` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`Codice`)
);

-- Una gestazione può esistere solo se il relativo tentativo
-- di riproduzione ha avuto successo (Vincolo generico n.3)
DELIMITER $$
DROP TRIGGER IF EXISTS `VerificaRiproduzione` $$
CREATE TRIGGER `VerificaRiproduzione` BEFORE INSERT ON `Gestazione` FOR EACH
ROW
BEGIN
  DECLARE successo TINYINT(1) DEFAULT 1; -- 0 (false) se la riproduzione
  è fallita

  SELECT IF(Stato = 'Successo', 1, 0) INTO successo
  FROM Riproduzione
  WHERE Istante = NEW.Istante
    AND Padre = NEW.Padre
    AND Madre = NEW.Madre;

  IF successo = 0 THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'La riproduzione è fallita e quindi non può
    esserci una gestazione.';
  END IF;
END $$

-- Controlla anche l'aggiornamento
DROP TRIGGER IF EXISTS `VerificaRiproduzione_U` $$
CREATE TRIGGER `VerificaRiproduzione_U` BEFORE UPDATE ON `Gestazione` FOR
EACH ROW
BEGIN
  DECLARE successo TINYINT(1) DEFAULT 1; -- 0 (false) se la riproduzione
  è fallita

  IF OLD.Istante <> NEW.Istante OR OLD.Padre <> NEW.Padre OR OLD.Madre <>
  NEW.Madre THEN
    SELECT IF(Stato = 'Successo', 1, 0) INTO successo
    FROM Riproduzione
    WHERE Istante = NEW.Istante
      AND Padre = NEW.Padre
      AND Madre = NEW.Madre;

    IF successo = 0 THEN
      SIGNAL SQLSTATE '45000'
      SET MESSAGE_TEXT = 'La riproduzione è fallita e quindi non
      può esserci una gestazione.';
    END IF;
  END IF;
END $$
DELIMITER ;

-- -----
-- Creazione tabella Controllo

```

```

-- -----
DROP TABLE IF EXISTS `Controllo`;
CREATE TABLE `Controllo` (
    `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
    `Stato` VARCHAR(15) NOT NULL, -- Programmato o Svolto
    `DataProgrammata` DATE NOT NULL,
    `DataEffettiva` DATE NULL,
    `Esito` VARCHAR(10) NULL, -- Positivo o Negativo
    `Gestazione` INT UNSIGNED NOT NULL,
    `Veterinario` INT UNSIGNED NOT NULL,
    PRIMARY KEY (`Codice`)
);

-- -----
-- Creazione tabella Esame
-- -----
DROP TABLE IF EXISTS `Esame`;
CREATE TABLE `Esame` (
    `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
    `Nome` VARCHAR(50) NOT NULL,
    `Data` DATE NOT NULL,
    `Macchinario` VARCHAR(30) NULL,
    `Descrizione` VARCHAR(1000) NOT NULL,
    `CodiceControllo` INT UNSIGNED NOT NULL,
    PRIMARY KEY (`Codice`)
);

-- -----
-- Creazione tabella Veterinario
-- -----
DROP TABLE IF EXISTS `Veterinario`;
CREATE TABLE `Veterinario` (
    `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
    `Nome` VARCHAR(40) NOT NULL,
    `Cognome` VARCHAR(40) NOT NULL,
    `Telefono` VARCHAR(15) NOT NULL,
    `Agriturismo` INT UNSIGNED NOT NULL,
    PRIMARY KEY (`Codice`)
);

-- -----
-- Creazione tabella Mungitrice
-- -----
DROP TABLE IF EXISTS `Mungitrice`;
CREATE TABLE `Mungitrice` (
    `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
    `Latitudine` DECIMAL(7, 5) NOT NULL, -- 7 = cifre totali massime, 5 =
cifre dopo la virgola che ci danno una precisione di 1 metro
    `Longitudine` DECIMAL(8, 5) NOT NULL, -- 8 = cifre totali massime, 5 =
cifre dopo la virgola che ci danno una precisione di 1 metro
    PRIMARY KEY (`Codice`)
);

-- -----
-- Creazione tabella Latte
-- -----
DROP TABLE IF EXISTS `Latte`;
CREATE TABLE `Latte` (
    `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
    `Volume` TINYINT UNSIGNED NOT NULL, -- In litri

```

```

        `Mungitrice` INT UNSIGNED NOT NULL,
        `Animale` INT UNSIGNED NOT NULL,
        `Silos` INT UNSIGNED NOT NULL,
        PRIMARY KEY (`Codice`)
    );

-- -----
-- Creazione tabella Caratteristica
-- -----
DROP TABLE IF EXISTS `Caratteristica`;
CREATE TABLE `Caratteristica` (
    `Sostanza` VARCHAR(50) NOT NULL,
    `CodiceLatte` INT UNSIGNED NOT NULL,
    `QuantitaMgMl` FLOAT UNSIGNED NOT NULL, -- Mg di sostanza contenuta in
1ml (Mg/ml)
    PRIMARY KEY (`Sostanza`, `CodiceLatte`)
);

-- -----
-- Creazione tabella Silos
-- -----
DROP TABLE IF EXISTS `Silos`;
CREATE TABLE `Silos` (
    `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
    `LivelloRiempimento` SMALLINT UNSIGNED NOT NULL, -- Litri di latte che
contiene il silos
    `CapacitaEttolitri` SMALLINT UNSIGNED NOT NULL,
    `Agriturismo` INT UNSIGNED NOT NULL,
    PRIMARY KEY (`Codice`)
);

-- -----
-- Creazione tabella LaboratorioProduzione
-- -----
DROP TABLE IF EXISTS `LaboratorioProduzione`;
CREATE TABLE `LaboratorioProduzione` (
    `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
    `Superficie` SMALLINT UNSIGNED NOT NULL, -- In m^2
    `NumeroVasche` TINYINT UNSIGNED NOT NULL,
    `Agriturismo` INT UNSIGNED NOT NULL,
    PRIMARY KEY (`Codice`)
);

-- -----
-- Creazione tabella Lotto
-- -----
DROP TABLE IF EXISTS `Lotto`;
CREATE TABLE `Lotto` (
    `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
    `Data` DATE NOT NULL,
    `Scadenza` DATE NOT NULL,
    `DurataProduzione` SMALLINT UNSIGNED NOT NULL, -- In minuti (tempi
stagionatura esclusi)
    `Silos` INT UNSIGNED NOT NULL,
    `Laboratorio` INT UNSIGNED NOT NULL,
    PRIMARY KEY (`Codice`)
);

-- -----
-- Creazione tabella ParametroEffettivo

```

```

-- -----
DROP TABLE IF EXISTS `ParametroEffettivo`;
CREATE TABLE `ParametroEffettivo` (
    -- Durata (della fase, in minuti), Tempi stagionatura (in gg),
    Temperatura di cottura (gradi),
    -- Temperatura del latte (gradi), Tempi di riposo (minuti), Temperatura
    caglio (gradi),
    -- Intervallo tra rottura (minuti)
    `Nome` VARCHAR(50) NOT NULL,
    `Fase` TINYINT UNSIGNED NOT NULL,
    `Lotto` INT UNSIGNED NOT NULL,
    `Valore` SMALLINT UNSIGNED NOT NULL, -- Gradi Celsius/MInuti/Giorni
    PRIMARY KEY (`Nome`, `Fase`, `Lotto`)
);

-- -----
-- Creazione tabella Operaio
-- -----
DROP TABLE IF EXISTS `Operaio`;
CREATE TABLE `Operaio` (
    `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
    `Stipendio` SMALLINT UNSIGNED NOT NULL,
    `Nome` VARCHAR(40) NOT NULL,
    `Cognome` VARCHAR(40) NOT NULL,
    PRIMARY KEY (`Codice`)
);

-- -----
-- Creazione tabella SchedaValutazione
-- -----
DROP TABLE IF EXISTS `SchedaValutazione`;
CREATE TABLE `SchedaValutazione` (
    `ProdottoCaseario` INT UNSIGNED NOT NULL,
    `Odore` VARCHAR(30) NOT NULL,
    `Colore` VARCHAR(30) NOT NULL,
    `Muffa` TINYINT(1) NOT NULL,
    `ValutazioneGenerale` TINYINT UNSIGNED NOT NULL, -- Da 1 a 10
    PRIMARY KEY (`ProdottoCaseario`)
);

-- -----
-- Creazione tabella ProdottoCaseario
-- -----
DROP TABLE IF EXISTS `ProdottoCaseario`;
CREATE TABLE `ProdottoCaseario` (
    `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
    `Peso` FLOAT UNSIGNED NOT NULL, -- In kg
    `Lotto` INT UNSIGNED NOT NULL,
    `Formaggio` VARCHAR(30) NOT NULL,
    `Scaffalatura` INT UNSIGNED NOT NULL,
    `Ordine` INT UNSIGNED NULL,
    `Sconto` TINYINT UNSIGNED NOT NULL DEFAULT 0,
    `FineStagionatura` DATE NULL, -- Null se non necessita di stagionatura
    PRIMARY KEY (`Codice`)
);

-- "Aggiornamento" della ridondanza 'FineStagionatura' (cap. 5.3.2) della
tabella ProdottoCaseario
DELIMITER $$
DROP TRIGGER IF EXISTS `AggiornamentoFineStagionatura` $$

```

```

CREATE TRIGGER `AggiornamentoFineStagionatura` BEFORE INSERT ON
`ProdottoCaseario` FOR EACH ROW
BEGIN
    DECLARE data_produzione DATE;
    DECLARE giorni_stagionatura SMALLINT UNSIGNED DEFAULT NULL;

    -- Se non deve essere stagionato tempi_stagionatura rimarrà NULL
    SELECT Valore INTO giorni_stagionatura
    FROM ParametroProcesso
    WHERE Formaggio = NEW.Formaggio AND Nome = 'Tempi stagionatura';

    SELECT `Data` INTO data_produzione
    FROM Lotto
    WHERE Codice = NEW.Lotto;

    IF giorni_stagionatura IS NOT NULL THEN
        SET NEW.FineStagionatura = data_produzione + INTERVAL
giorni_stagionatura DAY;
    END IF;
END $$
DELIMITER ;

-- (Vincolo generico n.11)
-- Quando viene aggiunto un ordine ad un prodotto caseario verifico se in
quell'ordine mancano ancora
-- altri formaggi della stessa tipologia (per esempio perchè non è stata
ancora raggiunta la quantità ordinata).
-- Nel caso non sia così (e quindi l'ordine, per quella tipologia di
formaggio, sia completo)
-- rimuovo il tipo del prodotto dalla tabella Attesa
DELIMITER $$
DROP TRIGGER IF EXISTS `ControlloAttesa` $$
CREATE TRIGGER `ControlloAttesa` AFTER INSERT ON `ProdottoCaseario` FOR EACH
ROW
BEGIN
    DECLARE quantita_rimanente FLOAT UNSIGNED; -- Quantità di prodotti
ancora in attesa

    -- Se ha già un ordine
    IF NEW.Ordine IS NOT NULL THEN

        -- Trovo la quantità del tipo del formaggio appena inserito in
attesa
        SELECT Quantita INTO quantita_rimanente
        FROM Attesa
        WHERE Formaggio = NEW.Formaggio
            AND Ordine = NEW.Ordine;

        IF quantita_rimanente - NEW.Peso <= 0 THEN
            DELETE FROM Attesa
            WHERE Ordine = NEW.Ordine AND Formaggio = NEW.Formaggio;

        ELSE
            UPDATE Attesa
            SET Quantita = quantita_rimanente - NEW.Peso
            WHERE Formaggio = NEW.Formaggio
                AND Ordine = NEW.Ordine;
        END IF;
    END IF;
END IF;

```

```

END $$

DROP TRIGGER IF EXISTS `ControlloAttesa_U` $$
CREATE TRIGGER `ControlloAttesa_U` AFTER UPDATE ON `ProdottoCaseario` FOR
EACH ROW
BEGIN
    DECLARE quantita_rimanente FLOAT UNSIGNED; -- Quantità di prodotti
ancora in attesa

    -- Se è stato inserito l'ordine
    IF OLD.Ordine IS NULL AND NEW.Ordine IS NOT NULL THEN

        -- Trovo la quantità del tipo del formaggio appena inserito in
attesa
        SELECT Quantita INTO quantita_rimanente
        FROM Attesa
        WHERE Formaggio = NEW.Formaggio
            AND Ordine = NEW.Ordine;

        IF quantita_rimanente - NEW.Peso <= 0 THEN
            DELETE FROM Attesa
            WHERE Ordine = NEW.Ordine AND Formaggio = NEW.Formaggio;

        ELSE
            UPDATE Attesa
            SET Quantita = quantita_rimanente - NEW.Peso
            WHERE Formaggio = NEW.Formaggio
                AND Ordine = NEW.Ordine;
        END IF;
    END IF;
END $$
DELIMITER ;

-- -----
-- Creazione tabella Formaggio
-- -----
DROP TABLE IF EXISTS `Formaggio`;
CREATE TABLE `Formaggio` (
    `Tipo` VARCHAR(30) NOT NULL,
    `ZonaOrigine` VARCHAR(40) NOT NULL,
    `TipoPasta` VARCHAR(10) NOT NULL, -- Molle o Dura
    `Deperibilita` VARCHAR(10) NOT NULL, -- Alta, Bassa o Media
    `Prezzo` FLOAT UNSIGNED NOT NULL, -- €/100g
    PRIMARY KEY (`Tipo`)
);

-- -----
-- Creazione tabella Fase
-- -----
DROP TABLE IF EXISTS `Fase`;
CREATE TABLE `Fase` (
    `Fase` TINYINT UNSIGNED NOT NULL,
    `Formaggio` VARCHAR(30) NOT NULL,
    `Descrizione` VARCHAR(1000) NOT NULL,
    PRIMARY KEY (`Fase`, `Formaggio`)
);

-- -----
-- Creazione tabella ParametroProcesso

```



```

-- -----
DROP TABLE IF EXISTS `ParametroProcesso`;
CREATE TABLE `ParametroProcesso` (
    -- Durata (della fase, in minuti), Tempi stagionatura (in gg),
    Temperatura di cottura (gradi),
    -- Temperatura del latte (gradi), Tempi di riposo (minuti), Temperatura
    caglio (gradi),
    -- Intervallo tra rottura (minuti)
    `Nome` VARCHAR(50) NOT NULL,
    `Fase` TINYINT UNSIGNED NOT NULL,
    `Formaggio` VARCHAR(30) NOT NULL,
    `Valore` SMALLINT UNSIGNED NOT NULL, -- Gradi Celsius/Minuti/Giorni
    PRIMARY KEY (`Nome`, `Fase`, `Formaggio`)
);

-- -----
-- Creazione tabella Magazzino
-- -----
DROP TABLE IF EXISTS `Magazzino`;
CREATE TABLE `Magazzino` (
    `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
    `Superficie` SMALLINT UNSIGNED NOT NULL, -- In m^2
    `ClasseEnergetica` VARCHAR(10) NOT NULL,
    `Agriturismo` INT UNSIGNED NOT NULL,
    PRIMARY KEY (`Codice`)
);

-- -----
-- Creazione tabella Cantina
-- -----
DROP TABLE IF EXISTS `Cantina`;
CREATE TABLE `Cantina` (
    `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
    `ClasseEnergetica` VARCHAR(10) NOT NULL,
    `Agriturismo` INT UNSIGNED NOT NULL,
    PRIMARY KEY (`Codice`)
);

-- -----
-- Creazione tabella Scaffalatura
-- -----
DROP TABLE IF EXISTS `Scaffalatura`;
CREATE TABLE `Scaffalatura` (
    `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
    `Marca` VARCHAR(20) NOT NULL,
    `Portata` SMALLINT UNSIGNED NOT NULL, -- Portata in kg di tutta la
    scaffalatura
    `CapacitaFormaggi` SMALLINT UNSIGNED NOT NULL, -- Numero massimo di
    formaggi sostenibile
    `Magazzino` INT UNSIGNED NULL,
    `Cantina` INT UNSIGNED NULL,
    PRIMARY KEY (`Codice`)
);

-- Controllo che una scaffalatura si trovi o in un magazzino o in una
-- cantina (Vincolo generico n.10)
DELIMITER $$
DROP TRIGGER IF EXISTS `ControlloPosizionamento` $$
CREATE TRIGGER `ControlloPosizionamento` BEFORE INSERT ON `Scaffalatura` FOR
EACH ROW

```

```

BEGIN
    -- Controllo che la scaffalatura sia stata collocata
    IF NEW.Magazzino IS NULL AND NEW.Cantina IS NULL THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Una scaffalatura deve trovarsi in un
magazzino o in una cantina!';
    END IF;

    -- Controllo che la scaffalatura non si trovi sia in un magazzino che
    in una cantina
    IF NEW.Cantina IS NOT NULL AND NEW.Magazzino IS NOT NULL THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Un scaffalatura non può trovarsi
contemporaneamente sia in un magazzino che in una cantina!';
    END IF;
END $$

-- Controllo anche l'aggiornamento
DROP TRIGGER IF EXISTS `ControlloPosizionamento_U` $$
CREATE TRIGGER `ControlloPosizionamento_U` BEFORE UPDATE ON `Scaffalatura`
FOR EACH ROW
BEGIN
    -- Controllo che la scaffalatura sia stata collocata
    IF NEW.Magazzino IS NULL AND NEW.Cantina IS NULL THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Una scaffalatura deve trovarsi in un
magazzino o in una cantina!';
    END IF;

    -- Controllo che la scaffalatura non si trovi sia in un magazzino che
    in una cantina
    IF NEW.Cantina IS NOT NULL AND NEW.Magazzino IS NOT NULL THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Un scaffalatura non può trovarsi
contemporaneamente sia in un magazzino che in una cantina!';
    END IF;
END $$
DELIMITER ;

-- -----
-- Creazione tabella StatoCantina
-- -----
DROP TABLE IF EXISTS `StatoCantina`;
CREATE TABLE `StatoCantina` (
    `Istante` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    `Cantina` INT UNSIGNED NOT NULL,
    `Ventilazione` VARCHAR(15) NOT NULL, -- Forte, Leggera o Spenta
    `Temperatura` TINYINT UNSIGNED NOT NULL, -- Gradi
    `Umidita` TINYINT UNSIGNED NOT NULL, -- % (da 0 a 100)
    PRIMARY KEY (`Istante`, `Cantina`)
);

-- -----
-- Creazione tabella Lavora
-- -----
DROP TABLE IF EXISTS `Lavora`;
CREATE TABLE `Lavora` (
    `Operaio` INT UNSIGNED NOT NULL,
    `Lotto` INT UNSIGNED NOT NULL,
    PRIMARY KEY (`Operaio`, `Lotto`)

```

```

);

-- -----
-- Creazione tabella Visita
-- -----
DROP TABLE IF EXISTS `Visita`;
CREATE TABLE `Visita` (
    `Istante` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    `Animale` INT UNSIGNED NOT NULL,
    `Veterinario` INT UNSIGNED NOT NULL,
    PRIMARY KEY (`Istante`, `Animale`)
);

-- -----
-- Creazione tabella Disturbo
-- -----
DROP TABLE IF EXISTS `Disturbo`;
CREATE TABLE `Disturbo` (
    `Nome` VARCHAR(50) NOT NULL,
    `IstanteVisita` TIMESTAMP NOT NULL,
    `Animale` INT UNSIGNED NOT NULL,
    `Entita` VARCHAR(10) NOT NULL, -- Lieve, Moderata o Grave
    PRIMARY KEY (`Nome`, `IstanteVisita`, `Animale`)
);

-- -----
-- Creazione tabella Lesione
-- -----
DROP TABLE IF EXISTS `Lesione`;
CREATE TABLE `Lesione` (
    `Tipologia` VARCHAR(50) NOT NULL,
    `ParteCorpo` VARCHAR(50) NOT NULL,
    `IstanteVisita` TIMESTAMP NOT NULL,
    `Animale` INT UNSIGNED NOT NULL,
    `Entita` VARCHAR(10) NOT NULL, -- Lieve, Moderata o Grave
    PRIMARY KEY (`Tipologia`, `ParteCorpo`, `IstanteVisita`, `Animale`)
);

-- -----
-- Creazione tabella Patologia
-- -----
DROP TABLE IF EXISTS `Patologia`;
CREATE TABLE `Patologia` (
    `Nome` VARCHAR(50) NOT NULL,
    `ParteCorpo` VARCHAR(50) NOT NULL,
    PRIMARY KEY (`Nome`)
);

-- -----
-- Creazione tabella Esordio
-- -----
DROP TABLE IF EXISTS `Esordio`;
CREATE TABLE `Esordio` (
    `Terapia` INT UNSIGNED NOT NULL,
    `IstanteVisita` TIMESTAMP NOT NULL,
    `Animale` INT UNSIGNED NOT NULL,
    `Patologia` VARCHAR(50) NOT NULL,
    `Gravita` TINYINT UNSIGNED NOT NULL, -- Da 1 a 10
    PRIMARY KEY (`Terapia`)
);

```

```

-- Controllo che la gravità della patologia sia compresa tra 1 e 10 (Vincolo
generico n.13)
DELIMITER $$
DROP TRIGGER IF EXISTS `ControlloGravita` $$
CREATE TRIGGER `ControlloGravita` BEFORE INSERT ON `Esordio` FOR EACH ROW
BEGIN
    IF NEW.Gravita NOT BETWEEN 1 AND 10 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'La gravità è una valore compreso tra 1 e
10!';
    END IF;
END $$

-- Controllo anche l'aggiornamento
DROP TRIGGER IF EXISTS `ControlloGravita_U` $$
CREATE TRIGGER `ControlloGravita_U` BEFORE UPDATE ON `Esordio` FOR EACH ROW
BEGIN
    IF NEW.Gravita NOT BETWEEN 1 AND 10 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'La gravità è una valore compreso tra 1 e
10!';
    END IF;
END $$
DELIMITER ;

-- -----
-- Creazione tabella Comporta
-- -----
DROP TABLE IF EXISTS `Comporta`;
CREATE TABLE `Comporta` (
    `Terapia` INT UNSIGNED NOT NULL,
    `Controllo` INT UNSIGNED NOT NULL,
    PRIMARY KEY (`Terapia`)
);

-- -----
-- Creazione tabella Terapia
-- -----
DROP TABLE IF EXISTS `Terapia`;
CREATE TABLE `Terapia` (
    `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
    `DataInizio` DATE NOT NULL,
    `Durata` SMALLINT UNSIGNED NOT NULL,
    `Animale` INT UNSIGNED NOT NULL,
    PRIMARY KEY (`Codice`)
);

-- -----
-- Creazione tabella Farmaco
-- -----
DROP TABLE IF EXISTS `Farmaco`;
CREATE TABLE `Farmaco` (
    `Nome` VARCHAR(50) NOT NULL,
    `Posologia` TINYINT UNSIGNED NOT NULL,
    `PrincipioAttivo` VARCHAR(50) NOT NULL,
    PRIMARY KEY (`Nome`)
);

-- -----

```

```

-- Creazione tabella OrarioSomministrazione
-- -----
DROP TABLE IF EXISTS `OrarioSomministrazione`;
CREATE TABLE `OrarioSomministrazione` (
  `Orario` TIME NOT NULL,
  `Farmaco` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`Orario`, `Farmaco`)
);

-- -----
-- Creazione tabella Somministra
-- -----
DROP TABLE IF EXISTS `Somministra`;
CREATE TABLE `Somministra` (
  `Terapia` INT UNSIGNED NOT NULL,
  `Farmaco` VARCHAR(50) NOT NULL,
  `GiorniPausa` TINYINT UNSIGNED NOT NULL,
  PRIMARY KEY (`Terapia`, `Farmaco`)
);

-- -----
-- Creazione tabella Camera
-- -----
DROP TABLE IF EXISTS `Camera`;
CREATE TABLE `Camera` (
  `Numero` SMALLINT UNSIGNED NOT NULL,
  `Agriturismo` INT UNSIGNED NOT NULL,
  `Tipo` VARCHAR(15) NOT NULL, -- Suite o Semplice
  `Costo` SMALLINT UNSIGNED NOT NULL,
  `DataArrivo` DATE NULL,
  PRIMARY KEY (`Numero`, `Agriturismo`)
);

-- -----
-- Creazione tabella Letto
-- -----
DROP TABLE IF EXISTS `Letto`;
CREATE TABLE `Letto` (
  `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
  `Tipo` VARCHAR(15) NOT NULL, -- Matrimoniale o Singolo
  `Camera` SMALLINT UNSIGNED NOT NULL,
  `Agriturismo` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`Codice`)
);

-- -----
-- Creazione tabella Capienza
-- -----
DROP TABLE IF EXISTS `Capienza`;
CREATE TABLE `Capienza` (
  `Tipo` VARCHAR(15) NOT NULL, -- Tipo di letto
  `Capienza` TINYINT UNSIGNED NOT NULL, -- Numero persone
  PRIMARY KEY (`Tipo`)
);

-- -----
-- Creazione tabella Cliente
-- -----
DROP TABLE IF EXISTS `Cliente`;
CREATE TABLE `Cliente` (

```

```

        `CodiceDocumento` VARCHAR(50) NOT NULL,
        `Nome` VARCHAR(40) NOT NULL,
        `Cognome` VARCHAR(40) NOT NULL,
        `Registrato` TINYINT(1) NOT NULL,
        `CodiceCarta` VARCHAR(30) NOT NULL,
        PRIMARY KEY (`CodiceDocumento`)
);

-- -----
-- Creazione tabella Account
-- -----
DROP TABLE IF EXISTS `Account`;
CREATE TABLE `Account` (
    `CodiceDocumento` VARCHAR(50) NOT NULL,
    `Username` VARCHAR(30) NOT NULL,
    `Password` VARCHAR(40) NOT NULL,
    `CodiceFiscale` VARCHAR(20) NULL,
    `Indirizzo` VARCHAR(50) NOT NULL,
    `Telefono` VARCHAR(15) NULL,
    `DataIscrizione` DATE NOT NULL,
    `DomandaDiRecupero` VARCHAR(50) NOT NULL,
    `RispostaDiRecupero` VARCHAR(100) NOT NULL,
    PRIMARY KEY (`CodiceDocumento`)
);

-- -----
-- Creazione tabella Documento
-- -----
DROP TABLE IF EXISTS `Documento`;
CREATE TABLE `Documento` (
    `CodiceDocumento` VARCHAR(50) NOT NULL,
    `Tipo` VARCHAR(30) NOT NULL,
    `Scadenza` DATE NOT NULL,
    `EnteRilascio` VARCHAR(40) NOT NULL,
    PRIMARY KEY (`CodiceDocumento`)
);

-- Verifico che il documento fornito non sia scaduto (Vincolo generico n.5)
DELIMITER $$
DROP TRIGGER IF EXISTS `ControlloScadenzaDocumento` $$
CREATE TRIGGER `ControlloScadenzaDocumento` BEFORE INSERT ON `Documento` FOR
EACH ROW
BEGIN
    IF NEW.Scadenza < CURRENT_DATE THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Il documento è scaduto!';
    END IF;
END $$

-- Controlla anche l'aggiornamento
DROP TRIGGER IF EXISTS `ControlloScadenzaDocumento_U` $$
CREATE TRIGGER `ControlloScadenzaDocumento_U` BEFORE UPDATE ON `Documento`
FOR EACH ROW
BEGIN
    IF NEW.Scadenza < CURRENT_DATE THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Il documento è scaduto!';
    END IF;
END $$
DELIMITER ;

```

```

-- -----
-- Creazione tabella Prenotazione
-- -----
DROP TABLE IF EXISTS `Prenotazione`;
CREATE TABLE `Prenotazione` (
  `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
  `DataPartenza` DATE NOT NULL,
  `DataArrivo` DATE NOT NULL,
  `Cliente` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`Codice`)
);

-- Verifico che la correttezza della data di arrivo e di partenza di una
prenotazione (Vincolo generico n.7)
DELIMITER $$
DROP TRIGGER IF EXISTS `ControlloDate` $$
CREATE TRIGGER `ControlloDate` BEFORE INSERT ON `Prenotazione` FOR EACH ROW
BEGIN
  IF NEW.DataArrivo > NEW.DataPartenza THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'La data di arrivo non può essere antecedente
a quella di partenza!';
  END IF;
END $$

-- Controllo anche l'aggiornamento
DROP TRIGGER IF EXISTS `ControlloDate_U` $$
CREATE TRIGGER `ControlloDate_U` BEFORE UPDATE ON `Prenotazione` FOR EACH
ROW
BEGIN
  IF NEW.DataArrivo > NEW.DataPartenza THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'La data di arrivo non può essere antecedente
a quella di partenza!';
  END IF;
END $$
DELIMITER ;

-- -----
-- Creazione tabella ServizioAggiuntivo
-- -----
DROP TABLE IF EXISTS `ServizioAggiuntivo`;
CREATE TABLE `ServizioAggiuntivo` (
  `Servizio` VARCHAR(50) NOT NULL,
  `Costo` SMALLINT UNSIGNED NOT NULL,
  PRIMARY KEY (`Servizio`)
);

-- -----
-- Creazione tabella Pagamento
-- -----
DROP TABLE IF EXISTS `Pagamento`;
CREATE TABLE `Pagamento` (
  `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
  `Istante` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `MetodoPagamento` VARCHAR(30) NOT NULL,
  `Importo` SMALLINT UNSIGNED NOT NULL,
  `Prenotazione` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`Codice`)

```

```

);

-- -----
-- Creazione tabella Escursione
-- -----
DROP TABLE IF EXISTS `Escursione`;
CREATE TABLE `Escursione` (
  `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
  `OrarioInizio` TIMESTAMP NOT NULL,
  `Costo` SMALLINT UNSIGNED NOT NULL,
  `Guida` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`Codice`)
);

-- -----
-- Creazione tabella Guida
-- -----
DROP TABLE IF EXISTS `Guida`;
CREATE TABLE `Guida` (
  `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
  `Nome` VARCHAR(40) NOT NULL,
  `Cognome` VARCHAR(40) NOT NULL,
  `Agriturismo` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`Codice`)
);

-- -----
-- Creazione tabella Step
-- -----
DROP TABLE IF EXISTS `Step`;
CREATE TABLE `Step` (
  `Numero` TINYINT UNSIGNED NOT NULL,
  `Escursione` INT UNSIGNED NOT NULL,
  `TempoSosta` TINYINT UNSIGNED NOT NULL, -- In minuti
  `AreaTenuta` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`Numero`, `Escursione`)
);

-- -----
-- Creazione tabella AreaTenuta
-- -----
DROP TABLE IF EXISTS `AreaTenuta`;
CREATE TABLE `AreaTenuta` (
  `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
  `Superficie` SMALLINT UNSIGNED NOT NULL, -- In m^2
  `Agriturismo` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`Codice`)
);

-- -----
-- Creazione tabella Assegnazione
-- -----
DROP TABLE IF EXISTS `Assegnazione`;
CREATE TABLE `Assegnazione` (
  `Prenotazione` INT UNSIGNED NOT NULL,
  `Camera` SMALLINT UNSIGNED NOT NULL,
  `Agriturismo` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`Prenotazione`, `Camera`, `Agriturismo`)
);

```



```

-- Aggiornamento della ridondanza 'Data arrivo' (cap. 5.3.1) della tabella
Camera
DELIMITER $$
DROP TRIGGER IF EXISTS `AggiornamentoDataArrivo` $$
CREATE TRIGGER `AggiornamentoDataArrivo` AFTER INSERT ON `Assegnazione` FOR
EACH ROW
BEGIN
    UPDATE Camera
    SET DataArrivo = (SELECT DataArrivo FROM Prenotazione WHERE Codice =
NEW.Prenotazione)
    WHERE Numero = NEW.Camera
        AND Agriturismo = NEW.Agriturismo;
END $$
DELIMITER ;

-- -----
-- Creazione tabella Richiede
-- -----
DROP TABLE IF EXISTS `Richiede`;
CREATE TABLE `Richiede` (
    `Prenotazione` INT UNSIGNED NOT NULL,
    `Servizio` VARCHAR(50) NOT NULL,
    `NumeroGiorni` TINYINT UNSIGNED NOT NULL,
    PRIMARY KEY (`Prenotazione`, `Servizio`)
);

-- -----
-- Creazione tabella Prenota
-- -----
DROP TABLE IF EXISTS `Prenota`;
CREATE TABLE `Prenota` (
    `Prenotazione` INT UNSIGNED NOT NULL,
    `Escursione` INT UNSIGNED NOT NULL,
    PRIMARY KEY (`Prenotazione`, `Escursione`)
);

-- -----
-- Creazione tabella Ordine
-- -----
DROP TABLE IF EXISTS `Ordine`;
CREATE TABLE `Ordine` (
    `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
    `Istante` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    `Stato` VARCHAR(20) NOT NULL, -- pendente, in processazione, in
preparazione, spedito, evaso
    `Cliente` VARCHAR(50) NOT NULL,
    PRIMARY KEY (`Codice`)
);

-- Per acquistare prodotti online un cliente deve fornire il codice fiscale,
il numero di telefono
-- e la tipologia, la scadenza e l'ente di rilascio di un documento
d'identità
-- (Vincolo generico n.4)
DELIMITER $$
DROP TRIGGER IF EXISTS `ControlloDatiRichiesti` $$
CREATE TRIGGER `ControlloDatiRichiesti` BEFORE INSERT ON `Ordine` FOR EACH
ROW
BEGIN
    -- Informazioni richieste per acquistare prodotti

```

```

    DECLARE codice_fiscale VARCHAR(20);
    DECLARE telefono VARCHAR(15);
    DECLARE tipo_documento VARCHAR(30);
    DECLARE scadenza_documento DATE;
    DECLARE ente_rilascio VARCHAR(40);

    SELECT A.CodiceFiscale, A.Telefono, D.Tipo, D.Scadenza, D.EnteRilascio
           INTO codice_fiscale, telefono, tipo_documento,
scadenza_documento, ente_rilascio
    FROM `Account` A
        NATURAL JOIN
        Documento D
    WHERE A.CodiceDocumento = NEW.Cliente;

    IF codice_fiscale IS NULL
        OR telefono IS NULL
        OR tipo_documento IS NULL
        OR scadenza_documento IS NULL
        OR ente_rilascio IS NULL THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Non sono presenti tutte le informazioni
richieste!';
    END IF;
END $$
DELIMITER ;

-- Controllo che gli ordini seguano la sequenza di stati (Vincolo generico
n.6)
DELIMITER $$
DROP TRIGGER IF EXISTS `ControlloStati` $$
CREATE TRIGGER `ControlloStati` BEFORE UPDATE ON `Ordine` FOR EACH ROW
BEGIN
    -- Se è stato aggiornato lo stato
    IF OLD.Stato <> NEW.Stato THEN

        IF NEW.Stato = 'Pendente' THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Un ordine non può tornare allo stato
pendente!';
        END IF;

        IF NEW.Stato = 'In processazione' AND OLD.Stato <> 'Pendente' THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Un ordine non può tornare allo stato
''in processazione''!';
        END IF;

        IF NEW.Stato = 'In preparazione' AND OLD.Stato <> 'In
processazione' THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Un ordine deve prima passare dallo
stato ''in processazione''!';
        END IF;

        IF NEW.Stato = 'Spedito' AND OLD.Stato <> 'In preparazione' THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Un ordine deve prima passare dallo
stato ''in preparazione''!';
        END IF;
    END IF;
END;

```

```

        IF NEW.Stato = 'Evaso' AND OLD.Stato <> 'Spedito' THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Un ordine deve prima passare dallo
stato ''spedito''!';
        END IF;
    END IF;
END $$
DELIMITER ;

-- -----
-- Creazione tabella Recensione
-- -----
DROP TABLE IF EXISTS `Recensione`;
CREATE TABLE `Recensione` (
    `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
    `Gradimento` TINYINT UNSIGNED NOT NULL, -- Da 1 a 5
    `Gusto` TINYINT UNSIGNED NOT NULL, -- Da 1 a 5
    `Qualita` TINYINT UNSIGNED NOT NULL, -- Da 1 a 5
    `Conservazione` TINYINT UNSIGNED NOT NULL, -- Da 1 a 5
    `NoteAggiuntive` VARCHAR(1000) NOT NULL DEFAULT '',
    `CodiceOrdine` INT UNSIGNED NOT NULL,
    PRIMARY KEY (`Codice`)
);

-- -----
-- Creazione tabella Spedizione
-- -----
DROP TABLE IF EXISTS `Spedizione`;
CREATE TABLE `Spedizione` (
    `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
    `Stato` VARCHAR(20) NOT NULL, -- Spedita, In transito, In consegna,
Consegnata
    `DataConsegnaPrevista` DATE NOT NULL,
    `DataConsegnaEffettiva` DATE NULL,
    `CodiceOrdine` INT UNSIGNED NOT NULL,
    PRIMARY KEY (`Codice`)
);

-- -----
-- Creazione tabella CentroSmistamento
-- -----
DROP TABLE IF EXISTS `CentroSmistamento`;
CREATE TABLE `CentroSmistamento` (
    `Codice` INT UNSIGNED AUTO_INCREMENT NOT NULL,
    `NumeroDipendenti` SMALLINT UNSIGNED NOT NULL,
    `Latitudine` DECIMAL(7, 5) NOT NULL, -- 7 = cifre totali massime, 5 =
cifre dopo la virgola che ci danno una precisione di 1 metro
    `Longitudine` DECIMAL(8, 5) NOT NULL, -- 8 = cifre totali massime, 5 =
cifre dopo la virgola che ci danno una precisione di 1 metro
    PRIMARY KEY (`Codice`)
);

-- -----
-- Creazione tabella Attesa
-- -----
DROP TABLE IF EXISTS `Attesa`;
CREATE TABLE `Attesa` (
    `Ordine` INT UNSIGNED NOT NULL,
    `Formaggio` VARCHAR(30) NOT NULL,
    `Quantita` FLOAT NOT NULL, -- In kg

```

```

        PRIMARY KEY (`Ordine`, `Formaggio`)
    );

-- Verifico che i grammi ordinati siano un multiplo di 100 (Vincolo generico
n.12)
DELIMITER $$
DROP TRIGGER IF EXISTS `ControlloQuantita` $$
CREATE TRIGGER `ControlloQuantita` BEFORE INSERT ON `Attesa` FOR EACH ROW
BEGIN
    -- Per qualche ragione la moltiplicazione non viene eseguita in modo
corretto/con priorità
    -- all'interno dell'operatore MOD, quindi è necessaria la variabile
    DECLARE grammi FLOAT UNSIGNED DEFAULT NEW.Quantita * 10;

    IF MOD(grammi, 1) <> 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'La quantità ordinata deve essere un multiplo
di 100g!';
    END IF;
END $$
DELIMITER ;

-- -----
-- Creazione tabella Percorso
-- -----
DROP TABLE IF EXISTS `Percorso`;
CREATE TABLE `Percorso` (
    `Hub` INT UNSIGNED NOT NULL,
    `Spedizione` INT UNSIGNED NOT NULL,
    PRIMARY KEY (`Hub`, `Spedizione`)
);

-- -----
-- Creazione tabella Monitoraggio
-- -----
DROP TABLE IF EXISTS `Monitoraggio`;
CREATE TABLE `Monitoraggio` (
    `IstanteVisita` TIMESTAMP NOT NULL,
    `Animale` INT UNSIGNED NOT NULL,
    `Indicatore` VARCHAR(50) NOT NULL,
    `Valore` VARCHAR(50) NOT NULL,
    PRIMARY KEY (`IstanteVisita`, `Animale`, `Indicatore`)
);

-- -----
-- Creazione tabella IndicatoreBenessere
-- -----
DROP TABLE IF EXISTS `IndicatoreBenessere`;
CREATE TABLE `IndicatoreBenessere` (
    `Nome` VARCHAR(50) NOT NULL,
    PRIMARY KEY (`Nome`)
);

-- -----
-- Foreign keys
-- -----
ALTER TABLE `Stalla`
    ADD FOREIGN KEY (`Agriturismo`) REFERENCES Agriturismo(`Codice`)
    ON UPDATE CASCADE ON DELETE RESTRICT;

```

```

ALTER TABLE `Locale`
    ADD FOREIGN KEY (`Stalla`) REFERENCES Stalla(`Codice`)
        ON UPDATE CASCADE ON DELETE RESTRICT,
    ADD FOREIGN KEY (`Specie`) REFERENCES Specie(`Nome`)
        ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `InterventoPulizia`
    ADD FOREIGN KEY (`Locale`) REFERENCES Locale(`Codice`)
        ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `Finestra`
    ADD FOREIGN KEY (`Locale`) REFERENCES Locale(`Codice`)
        ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `StatoAria`
    ADD FOREIGN KEY (`Locale`) REFERENCES Locale(`Codice`)
        ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `Mangiatoia`
    ADD FOREIGN KEY (`Locale`) REFERENCES Locale(`Codice`)
        ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `Abbeveratoio`
    ADD FOREIGN KEY (`Locale`) REFERENCES Locale(`Codice`)
        ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `Condizionatore`
    ADD FOREIGN KEY (`Locale`) REFERENCES Locale(`Codice`)
        ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `DispositivoIlluminazione`
    ADD FOREIGN KEY (`Locale`) REFERENCES Locale(`Codice`)
        ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `StatoMangiatoia`
    ADD FOREIGN KEY (`Mangiatoia`) REFERENCES Mangiatoia(`Codice`)
        ON UPDATE CASCADE ON DELETE CASCADE,
    ADD FOREIGN KEY (`Foraggio`) REFERENCES Foraggio(`Tipo`)
        ON UPDATE CASCADE ON DELETE SET NULL;

ALTER TABLE `StatoAbbeveratoio`
    ADD FOREIGN KEY (`Abbeveratoio`) REFERENCES Abbeveratoio(`Codice`)
        ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `ProprietaAcqua`
    ADD FOREIGN KEY (`Istante`, `Abbeveratoio`) REFERENCES
StatoAbbeveratoio(`Istante`, `Abbeveratoio`)
        ON UPDATE CASCADE ON DELETE CASCADE,
    ADD FOREIGN KEY (`Sostanza`) REFERENCES SostanzaChimica(`Nome`)
        ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `Animale`
    ADD FOREIGN KEY (`Locale`) REFERENCES Locale(`Codice`)
        ON UPDATE CASCADE ON DELETE RESTRICT,
    ADD FOREIGN KEY (`Razza`) REFERENCES Razza(`Nome`)
        ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `Razza`
    ADD FOREIGN KEY (`Specie`) REFERENCES Specie(`Nome`)
        ON UPDATE CASCADE ON DELETE RESTRICT;

```

```

ALTER TABLE `ZonaPascolo`
  ADD FOREIGN KEY (`Agriturismo`) REFERENCES Agriturismo(`Codice`)
    ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `Posizione`
  ADD FOREIGN KEY (`Animale`) REFERENCES Animale(`Codice`)
    ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `Recinzione`
  ADD FOREIGN KEY (`ZonaPascolo`, `Agriturismo`) REFERENCES
ZonaPascolo(`Nome`, `Agriturismo`)
    ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `Riproduzione`
  ADD FOREIGN KEY (`Padre`) REFERENCES Animale(`Codice`)
    ON UPDATE CASCADE ON DELETE RESTRICT,
  ADD FOREIGN KEY (`Madre`) REFERENCES Animale(`Codice`)
    ON UPDATE CASCADE ON DELETE RESTRICT,
  ADD FOREIGN KEY (`Supervisore`) REFERENCES Veterinario(`Codice`)
    ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `Gestazione`
  ADD FOREIGN KEY (`Istante`, `Padre`, `Madre`) REFERENCES
Riproduzione(`Istante`, `Padre`, `Madre`)
    ON UPDATE CASCADE ON DELETE CASCADE,
  ADD FOREIGN KEY (`Responsabile`) REFERENCES Veterinario(`Codice`)
    ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `Complicanza`
  ADD FOREIGN KEY (`Gestazione`) REFERENCES Gestazione(`Codice`)
    ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `Controllo`
  ADD FOREIGN KEY (`Gestazione`) REFERENCES Gestazione(`Codice`)
    ON UPDATE CASCADE ON DELETE CASCADE,
  ADD FOREIGN KEY (`Veterinario`) REFERENCES Veterinario(`Codice`)
    ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `Esame`
  ADD FOREIGN KEY (`CodiceControllo`) REFERENCES Controllo(`Codice`)
    ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `Veterinario`
  ADD FOREIGN KEY (`Agriturismo`) REFERENCES Agriturismo(`Codice`)
    ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `Latte`
  ADD FOREIGN KEY (`Animale`) REFERENCES Animale(`Codice`)
    ON UPDATE CASCADE ON DELETE CASCADE,
  ADD FOREIGN KEY (`Mungitrice`) REFERENCES Mungitrice(`Codice`)
    ON UPDATE CASCADE ON DELETE RESTRICT,
  ADD FOREIGN KEY (`Silos`) REFERENCES Silos(`Codice`)
    ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `Caratteristica`
  ADD FOREIGN KEY (`CodiceLatte`) REFERENCES Latte(`Codice`)
    ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `Silos`

```

```

        ADD FOREIGN KEY (`Agriturismo`) REFERENCES Agriturismo(`Codice`)
            ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `LaboratorioProduzione`
    ADD FOREIGN KEY (`Agriturismo`) REFERENCES Agriturismo(`Codice`)
        ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `Lotto`
    ADD FOREIGN KEY (`Silos`) REFERENCES Silos(`Codice`)
        ON UPDATE CASCADE ON DELETE RESTRICT,
    ADD FOREIGN KEY (`Laboratorio`) REFERENCES
LaboratorioProduzione(`Codice`)
        ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `ParametroEffettivo`
    ADD FOREIGN KEY (`Lotto`) REFERENCES Lotto(`Codice`)
        ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `ProdottoCaseario`
    ADD FOREIGN KEY (`Lotto`) REFERENCES Lotto(`Codice`)
        ON UPDATE CASCADE ON DELETE CASCADE,
    ADD FOREIGN KEY (`Formaggio`) REFERENCES Formaggio(`Tipo`)
        ON UPDATE CASCADE ON DELETE RESTRICT,
    ADD FOREIGN KEY (`Scaffalatura`) REFERENCES Scaffalatura(`Codice`)
        ON UPDATE CASCADE ON DELETE RESTRICT,
    ADD FOREIGN KEY (`Ordine`) REFERENCES Ordine(`Codice`)
        ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `SchedaValutazione`
    ADD FOREIGN KEY (`ProdottoCaseario`) REFERENCES
ProdottoCaseario(`Codice`)
        ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `Fase`
    ADD FOREIGN KEY (`Formaggio`) REFERENCES Formaggio(`Tipo`)
        ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `ParametroProcesso`
    ADD FOREIGN KEY (`Fase`, `Formaggio`) REFERENCES Fase(`Fase`,
`Formaggio`)
        ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `Magazzino`
    ADD FOREIGN KEY (`Agriturismo`) REFERENCES Agriturismo(`Codice`)
        ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `Cantina`
    ADD FOREIGN KEY (`Agriturismo`) REFERENCES Agriturismo(`Codice`)
        ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `Scaffalatura`
    ADD FOREIGN KEY (`Magazzino`) REFERENCES Magazzino(`Codice`)
        ON UPDATE CASCADE ON DELETE RESTRICT,
    ADD FOREIGN KEY (`Cantina`) REFERENCES Cantina(`Codice`)
        ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `StatoCantina`
    ADD FOREIGN KEY (`Cantina`) REFERENCES Cantina(`Codice`)
        ON UPDATE CASCADE ON DELETE CASCADE;

```

```

ALTER TABLE `Lavora`
    ADD FOREIGN KEY (`Operaio`) REFERENCES Operaio(`Codice`)
        ON UPDATE CASCADE ON DELETE RESTRICT,
    ADD FOREIGN KEY (`Lotto`) REFERENCES Lotto(`Codice`)
        ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `Visita`
    ADD FOREIGN KEY (`Veterinario`) REFERENCES Veterinario(`Codice`)
        ON UPDATE CASCADE ON DELETE RESTRICT,
    ADD FOREIGN KEY (`Animale`) REFERENCES Animale(`Codice`)
        ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `Disturbo`
    ADD FOREIGN KEY (`IstanteVisita`, `Animale`) REFERENCES
Visita(`Istante`, `Animale`)
        ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `Lesione`
    ADD FOREIGN KEY (`IstanteVisita`, `Animale`) REFERENCES
Visita(`Istante`, `Animale`)
        ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `Terapia`
    ADD FOREIGN KEY (`Animale`) REFERENCES Animale(`Codice`)
        ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `Comporta`
    ADD FOREIGN KEY (`Terapia`) REFERENCES Terapia(`Codice`)
        ON UPDATE CASCADE ON DELETE CASCADE,
    ADD FOREIGN KEY (`Controllo`) REFERENCES Controllo(`Codice`)
        ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `Esordio`
    ADD FOREIGN KEY (`Terapia`) REFERENCES Terapia(`Codice`)
        ON UPDATE CASCADE ON DELETE CASCADE,
    ADD FOREIGN KEY (`IstanteVisita`, `Animale`) REFERENCES
Visita(`Istante`, `Animale`)
        ON UPDATE CASCADE ON DELETE CASCADE,
    ADD FOREIGN KEY (`Patologia`) REFERENCES Patologia(`Nome`)
        ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `OrarioSomministrazione`
    ADD FOREIGN KEY (`Farmaco`) REFERENCES Farmaco(`Nome`)
        ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `Somministra`
    ADD FOREIGN KEY (`Terapia`) REFERENCES Terapia(`Codice`)
        ON UPDATE CASCADE ON DELETE CASCADE,
    ADD FOREIGN KEY (`Farmaco`) REFERENCES Farmaco(`Nome`)
        ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `Camera`
    ADD FOREIGN KEY (`Agriturismo`) REFERENCES Agriturismo(`Codice`)
        ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `Letto`
    ADD FOREIGN KEY (`Camera`, `Agriturismo`) REFERENCES Camera(`Numero`,
`Agriturismo`)
        ON UPDATE CASCADE ON DELETE RESTRICT,
    ADD FOREIGN KEY (`Tipo`) REFERENCES Capienza(`Tipo`)

```



```

        ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `Account`
    ADD FOREIGN KEY (`CodiceDocumento`) REFERENCES
Cliente(`CodiceDocumento`)
        ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `Prenotazione`
    ADD FOREIGN KEY (`Cliente`) REFERENCES Cliente(`CodiceDocumento`)
        ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `Pagamento`
    ADD FOREIGN KEY (`Prenotazione`) REFERENCES Prenotazione(`Codice`)
        ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `Escursione`
    ADD FOREIGN KEY (`Guida`) REFERENCES Guida(`Codice`)
        ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `Guida`
    ADD FOREIGN KEY (`Agriturismo`) REFERENCES Agriturismo(`Codice`)
        ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `Step`
    ADD FOREIGN KEY (`Escursione`) REFERENCES Escursione(`Codice`)
        ON UPDATE CASCADE ON DELETE CASCADE,
    ADD FOREIGN KEY (`AreaTenuta`) REFERENCES AreaTenuta(`Codice`)
        ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `AreaTenuta`
    ADD FOREIGN KEY (`Agriturismo`) REFERENCES Agriturismo(`Codice`)
        ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `Assegnazione`
    ADD FOREIGN KEY (`Prenotazione`) REFERENCES Prenotazione(`Codice`)
        ON UPDATE CASCADE ON DELETE CASCADE,
    ADD FOREIGN KEY (`Camera`, `Agriturismo`) REFERENCES Camera(`Numero`,
`Agriturismo`)
        ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `Richiede`
    ADD FOREIGN KEY (`Prenotazione`) REFERENCES Prenotazione(`Codice`)
        ON UPDATE CASCADE ON DELETE CASCADE,
    ADD FOREIGN KEY (`Servizio`) REFERENCES ServizioAggiuntivo(`Servizio`)
        ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `Prenota`
    ADD FOREIGN KEY (`Prenotazione`) REFERENCES Prenotazione(`Codice`)
        ON UPDATE CASCADE ON DELETE CASCADE,
    ADD FOREIGN KEY (`Escursione`) REFERENCES Escursione(`Codice`)
        ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `Ordine`
    ADD FOREIGN KEY (`Cliente`) REFERENCES `Account`(`CodiceDocumento`)
        ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `Recensione`
    ADD FOREIGN KEY (`CodiceOrdine`) REFERENCES Ordine(`Codice`)
        ON UPDATE CASCADE ON DELETE CASCADE;

```

```

ALTER TABLE `Spedizione`
  ADD FOREIGN KEY (`CodiceOrdine`) REFERENCES Ordine(`Codice`)
    ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `Attesa`
  ADD FOREIGN KEY (`Ordine`) REFERENCES Ordine(`Codice`)
    ON UPDATE CASCADE ON DELETE CASCADE,
  ADD FOREIGN KEY (`Formaggio`) REFERENCES Formaggio(`Tipo`)
    ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `Percorso`
  ADD FOREIGN KEY (`Hub`) REFERENCES CentroSmistamento(`Codice`)
    ON UPDATE CASCADE ON DELETE RESTRICT,
  ADD FOREIGN KEY (`Spedizione`) REFERENCES Spedizione(`Codice`)
    ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `Monitoraggio`
  ADD FOREIGN KEY (`IstanteVisita`, `Animale`) REFERENCES
Visita(`Istante`, `Animale`)
    ON UPDATE CASCADE ON DELETE CASCADE,
  ADD FOREIGN KEY (`Indicatore`) REFERENCES IndicatoreBenessere(`Nome`)
    ON UPDATE CASCADE ON DELETE RESTRICT;

ALTER TABLE `Documento`
  ADD FOREIGN KEY (`CodiceDocumento`) REFERENCES
`Account`(`CodiceDocumento`)
    ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `Acquisto`
  ADD FOREIGN KEY (`Fornitore`) REFERENCES Fornitore(`PartitaIVA`)
    ON UPDATE CASCADE ON DELETE RESTRICT,
  ADD FOREIGN KEY (`Animale`) REFERENCES Animale(`Codice`)
    ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE `Nascita`
  ADD FOREIGN KEY (`Animale`) REFERENCES Animale(`Codice`)
    ON UPDATE CASCADE ON DELETE RESTRICT,
  ADD FOREIGN KEY (`Gestazione`) REFERENCES Gestazione(`Codice`)
    ON UPDATE CASCADE ON DELETE CASCADE;

-- -----
--      Operazioni significative      --
--              e                      --
--      Funzionalità back-end         --
-- -----
DELIMITER $$

-- Operazione 5.2.1: ottengo le camere prenotate per il giorno successivo
DROP PROCEDURE IF EXISTS CamereDaPreparare $$
CREATE PROCEDURE CamereDaPreparare(IN agriturismo INT UNSIGNED)
BEGIN
  CREATE TEMPORARY TABLE IF NOT EXISTS _CamereDaPreparare (
    Camera SMALLINT UNSIGNED NOT NULL,
    Prenotazione INT UNSIGNED NOT NULL,
    PRIMARY KEY(Camera, Prenotazione)
  );

  TRUNCATE TABLE _CamereDaPreparare;

  INSERT INTO _CamereDaPreparare

```

```

SELECT A.Camera, A.Prenotazione
FROM Assegnazione A
      INNER JOIN
      Prenotazione P ON P.Codice = A.Prenotazione
WHERE A.Agriturismo = agriturismo
      AND P.DataArrivo = CURRENT_DATE + INTERVAL 1 DAY; -- Se
l'arrivo è previsto per domani
END $$

```

```

-- Operazione 5.2.2 e funzionalità di back-end: sposta un formaggio in
magazzino se ha terminato la stagionatura
-- cercando di posizionare sullo stesso scaffale i prodotti di uno stesso
lotto (in modo da ottimizzare le operazioni di vendita)

```

```

DROP EVENT IF EXISTS `Stagionatura` $$
CREATE EVENT `Stagionatura` ON SCHEDULE EVERY 1 DAY
STARTS '2019-01-01 02:00:00'
DO
BEGIN
    DECLARE formaggio INT UNSIGNED; -- Codice del formaggio
    DECLARE agriturismo INT UNSIGNED; -- Agriturismo in cui si trova il
formaggio
    DECLARE peso FLOAT UNSIGNED; -- Peso del formaggio
    DECLARE scaffalatura INT UNSIGNED DEFAULT NULL; -- Codice scaffalatura
libera
    DECLARE finito TINYINT(1) DEFAULT 0; -- 1 quando i formaggi che devono
essere spostati sono finiti

```

```

    -- Prendo tutti i formaggi che sono in cantina ma hanno terminato la
stagionatura

```

```

    DECLARE formaggi_cur CURSOR FOR
    SELECT PC.Codice, PC.Peso, C.Agriturismo
    FROM Scaffalatura S
        INNER JOIN
        Cantina C ON C.Codice = S.Cantina
        INNER JOIN
        ProdottoCaseario PC ON S.Codice = PC.Scaffalatura
        INNER JOIN
        Lotto L ON L.Codice = PC.Lotto
    WHERE PC.FineStagionatura <= CURRENT_DATE
    ORDER BY L.Codice; -- In questo modo quando scorrerò il cursore avrò
già i prodotti di uno stesso lotto vicini

```

```

    DECLARE CONTINUE HANDLER FOR NOT FOUND
    SET finito = 1;

```

```

    -- Trovo uno scaffale per ogni prodotto (cercando di mantenere su uno
stesso scaffale i prodotti di uno stesso lotto per ottimizzare le operazioni
di vendita)

```

```

    OPEN formaggi_cur;
scan: LOOP
    FETCH formaggi_cur INTO formaggio, peso, agriturismo;
    IF finito = 1 THEN
        LEAVE scan;
    END IF;

```

```

    -- Cerco una scaffalatura libera nei magazzini dello stesso
agriturismo per collocarci i prodotti
    SELECT S.Codice INTO scaffalatura
    FROM Scaffalatura S

```

```

INNER JOIN
Magazzino M ON M.Codice = S.Magazzino
WHERE M.Agriturismo = agriturismo
-- Se sulla scaffalatura c'è posto per almeno un altro
formaggio
AND S.CapacitaFormaggi > (
SELECT COUNT(*)
FROM
ProdottoCaseario PC
LEFT OUTER
JOIN
Ordine O ON
PC.Ordine = O.Codice
WHERE
PC.Scaffalatura = S.Codice
-- Considera i
prodotti non venduti o non ancora spediti
AND (PC.Ordine
IS NULL OR O.Stato <> 'spedito' AND O.Stato <> 'evaso')
)
-- Se riesce a sostenere il peso con il nuovo formaggio
AND S.Portata >= peso + (
SELECT
FROM
ProdottoCaseario PC
LEFT OUTER
JOIN
Ordine O ON
PC.Ordine = O.Codice
WHERE
PC.Scaffalatura = S.Codice
-- Considera i
prodotti non venduti o non ancora spediti
AND (PC.Ordine
IS NULL OR O.Stato <> 'spedito' AND O.Stato <> 'evaso')
)
ORDER BY S.Codice -- In questo modo riempirò sempre la stessa
scaffalatura, per quanto possibile
LIMIT 1;

-- Se c'è una scaffalatura in magazzino libera
-- (non esco dal loop perchè è possibile che un formaggio più
leggero "trovi posto")
IF scaffalatura IS NOT NULL THEN

-- Sposto il formaggio che ha terminato la stagionatura
UPDATE ProdottoCaseario PC
SET PC.Scaffalatura = scaffalatura
WHERE PC.Codice = formaggio;

-- Notifica della disponibilità del nuovo prodotto in modo che
possa essere venduto
CALL NotificaDisponibilita(formaggio);
END IF;

END LOOP;
CLOSE formaggi_cur;
END $$

```

```

-- Operazione 5.2.3: conto le vendite e i resi di una tipologia di formaggio
DROP PROCEDURE IF EXISTS CalcolaVendite $$
CREATE PROCEDURE CalcolaVendite(IN tipologia_formaggio VARCHAR(50))
BEGIN
    DECLARE vendite INT UNSIGNED;
    DECLARE resi INT UNSIGNED;
    DECLARE finito TINYINT(1) DEFAULT 0;

    DECLARE cur CURSOR FOR
        WITH prodotti AS (
            SELECT PC.*, O.Istante
            FROM ProdottoCaseario PC
            INNER JOIN
                Ordine O ON O.Codice = PC.Ordine
            WHERE PC.Formaggio = tipologia_formaggio
            AND DATE(O.Istante) > CURRENT_DATE - INTERVAL 7 DAY --
Ordini effettuati entro una settimana
        )
        SELECT COUNT(IF(P.Ordine IS NOT NULL, 1, NULL)), IFNULL((
            SELECT COUNT(*)
            FROM prodotti P2
            INNER JOIN
                SchedaValutazione SV ON SV.ProdottoCaseario = P2.Codice
            WHERE P.Codice = P2.Codice
        ), 0) AS Resi
        FROM prodotti P;

    DECLARE CONTINUE HANDLER FOR NOT FOUND
    SET finito = 1;

    CREATE TEMPORARY TABLE IF NOT EXISTS _VenditeResi (
        Vendite INT UNSIGNED NOT NULL,
        Resi INT UNSIGNED NOT NULL
    );

    TRUNCATE TABLE _VenditeResi;

    OPEN cur;

scan: LOOP
    FETCH cur INTO vendite, resi;
    IF finito = 1 THEN
        LEAVE scan;
    END IF;

    INSERT INTO _VenditeResi VALUES
        (vendite, resi);
END LOOP;

CLOSE cur;
END $$

```

```

-- Operazione 5.2.4: inserisco la posizione rilevata dal sensore
DROP PROCEDURE IF EXISTS InserisciPosizione $$
CREATE PROCEDURE InserisciPosizione(IN codice_animale INT UNSIGNED,
                                     IN latitudine DECIMAL(7,
5),
                                     IN longitudine DECIMAL(8,
5))
BEGIN
    INSERT INTO Posizione (Animale, Latitudine, Longitudine)
    VALUES (codice_animale, latitudine, longitudine);
END $$

-- Operazione 5.2.5: calcolo il saldo del soggiorno che dovrà pagare il
cliente
DROP PROCEDURE IF EXISTS SaldoSoggiorno $$
CREATE PROCEDURE SaldoSoggiorno(IN codice_cliente VARCHAR(50),
                                INOUT saldo SMALLINT UNSIGNED)
BEGIN
    DECLARE prenotazione INT UNSIGNED DEFAULT NULL;

    -- Trovo il soggiorno attuale
    SELECT Codice INTO prenotazione
    FROM Prenotazione
    WHERE Cliente = codice_cliente
        AND CURRENT_DATE BETWEEN DataArrivo AND DataPartenza;

    SET saldo = 0;

    -- Se è stata trovata la prenotazione
    IF prenotazione IS NOT NULL THEN

        -- Costo delle camere
        SET saldo = saldo + (

            SELECT IFNULL(SUM(C.Costo), 0)
            FROM Assegnazione A
                 INNER JOIN
                 Camera C ON A.Camera =

C.Numero

            WHERE A.Prenotazione =

prenotazione

        );

        -- Servizi aggiuntivi
        SET saldo = saldo + (

            SELECT IFNULL(SUM(SA.Costo *

R.NumeroGiorni), 0)

            FROM Richiede R
                 NATURAL JOIN
                 ServizioAggiuntivo SA
            WHERE R.Prenotazione =

prenotazione

        );

        -- Escursioni
        SET saldo = saldo + (

            SELECT IFNULL(SUM(E.Costo), 0)
            FROM Prenota P
                 INNER JOIN

```

```

                                Escursione E ON E.Codice
= P.Escursione
                                WHERE P.Prenotazione =
prenotazione
                                );

                                -- Pagamenti già effettuati
                                SET saldo = saldo - (
                                SELECT IFNULL(SUM(P.Importo),
0)
                                FROM Pagamento P
                                WHERE P.Prenotazione =
prenotazione
                                );

                                END IF;
END $$

-- Operazione 5.2.6: assegno un prodotto caseario appena prodotto o che ha
terminato la stagionatura ad un ordine pendente
DROP PROCEDURE IF EXISTS NotificaDisponibilita $$
CREATE PROCEDURE NotificaDisponibilita(IN codice_prodotto INT UNSIGNED)
BEGIN
    -- Informazioni sul prodotto: tipologia e peso
    DECLARE tipo_formaggio VARCHAR(30) DEFAULT (SELECT Formaggio
FROM
ProdottoCaseario
WHERE
Codice = codice_prodotto);
    DECLARE peso FLOAT UNSIGNED DEFAULT (SELECT PC.Peso
FROM
ProdottoCaseario PC
WHERE PC.Codice =
codice_prodotto);

    DECLARE codice_ordine INT UNSIGNED DEFAULT (SELECT Ordine
FROM
ProdottoCaseario
WHERE
Codice = codice_prodotto);
    DECLARE quantita_rimanente FLOAT UNSIGNED; -- Quantità di formaggio
mancante

    -- Se il prodotto non è già stato aggiunto ad un ordine
    IF codice_ordine IS NULL THEN

        -- Ottengo un ordine a cui manca la tipologia e la quantità di
formaggio del prodotto dato in input
        SELECT Ordine, Quantita INTO codice_ordine, quantita_rimanente
FROM Attesa
WHERE Formaggio = tipo_formaggio
        -- Arrotonda per difetto i grammi (1,417 kg -> 1,4 kg -
22,799 kg -> 22,7 kg).
        -- In questo modo non rimarranno mai ordini con meno di
100g (dimensione minima dei prodotti) in attesa
        -- e la quantità spedita non sarà mai inferiore alla quantità
ordinata ma al massimo eccederà di 99 grammi
        AND Quantita >= (FLOOR(peso * 10) / 10)
        LIMIT 1;

```

```

-- Se c'è almeno un ordine a cui manca la tipologia e la quantità di
formaggio del prodotto dato in input
    IF quantita_rimanente IS NOT NULL THEN

        -- Inserisco il formaggio nella coda di spedizione
        UPDATE ProdottoCaseario
        SET Ordine = codice_ordine
        WHERE Codice = codice_prodotto;

        -- Controllo se l'ordine è completo
        IF (SELECT COUNT(*) FROM Attesa WHERE Ordine =
codice_ordine) = 0 THEN

            -- Aggiorno lo stato dell'ordine
            UPDATE Ordine
            SET Stato = 'In processazione'
            WHERE Codice = codice_ordine;
        END IF;
    END IF;
END IF;
END $$

-- Operazione 5.2.7: inserisco un nuovo tentativo di riproduzione
DROP PROCEDURE IF EXISTS InserisciRiproduzione $$
CREATE PROCEDURE InserisciRiproduzione(IN codice_madre VARCHAR(50),
                                       IN codice_padre
                                       VARCHAR(50),
                                       IN codice_vedeterinario
                                       VARCHAR(50))
BEGIN
    INSERT INTO Riproduzione
    VALUES (CURRENT_TIMESTAMP, codice_madre, codice_padre, 'In attesa',
codice_veterinario);
END $$

-- Operazione 5.2.8: ottengo gli animali sottoposti a terapie all'interno
dell'agriturismo specificato
DROP PROCEDURE IF EXISTS AnimaliInCura $$
CREATE PROCEDURE AnimaliInCura(IN agriturismo INT UNSIGNED)
BEGIN
    CREATE TEMPORARY TABLE IF NOT EXISTS _AnimaliInCura (
        Animale INT UNSIGNED NOT NULL,
        PRIMARY KEY (Animale)
    );

    TRUNCATE TABLE _AnimaliInCura;

    INSERT INTO _AnimaliInCura
    SELECT D.Animale
    FROM (
        -- Ottengo gli animali attualmente sottoposti terapie
        SELECT Animale
        FROM Terapia
        WHERE DataInizio + INTERVAL Durata DAY > CURRENT_DATE
    ) AS D
    INNER JOIN
    Animale A ON A.Codice = D.Animale

```



```

        INNER JOIN
        Locale L ON L.Codice = A.Locale
        INNER JOIN
        Stalla S ON S.Codice = L.Stalla
    WHERE S.Agriturismo = agriturismo; -- Escludo tutti gli animali
    sottoposti a terapie che non fanno parte dell'agriturismo specificato
END $$

```

```

-- Operazione 5.2.9: inserisco una nuova visita ed aggiorno la scheda
dell'animale

```

```

DROP PROCEDURE IF EXISTS InserisciVisita $$
CREATE PROCEDURE InserisciVisita(IN codice_animale INT UNSIGNED,
                                IN altezza TINYINT
                                UNSIGNED,
                                IN peso SMALLINT
                                UNSIGNED,
                                IN codice_veterinario INT UNSIGNED)
BEGIN
    INSERT INTO Visita (Veterinario, Animale)
    VALUES (codice_veterinario, codice_animale);

    UPDATE Animale
    SET Altezza = altezza, Peso = peso
    WHERE Codice = codice_animale;
END $$

```

```

-- Funzionalità di back-end: un animale deve essere spostato in quarantena
(per evitare fenomeni di contagio)

```

```

-- dopo che due terapie consecutive relative alla stessa patologia
falliscono

```

```

DROP TRIGGER IF EXISTS `Quarantena` $$
CREATE TRIGGER `Quarantena` AFTER INSERT ON `Esordio` FOR EACH ROW
BEGIN
    DECLARE data_inizio DATE; -- Data inizio terapia
    DECLARE terapie_fallite INTEGER UNSIGNED DEFAULT 0; -- Numero terapie
precedenti fallite
    DECLARE finito TINYINT(1) DEFAULT 0;

```

```

    -- Ottengo le terapie, relative alla stessa patologia e allo stesso
animale,

```

```

    -- concludo il giorno in cui è iniziata la terapia appena inserita
    -- indicando un fallimento della terapia precedente

```

```

    DECLARE cur CURSOR FOR

```

```

    SELECT T.DataInizio

```

```

    FROM Terapia T

```

```

        INNER JOIN

```

```

        Esordio E ON E.Terapia = T.Codice -- Terapie provenienti dalle
visite

```

```

    -- Terminate quando la terapia relativa all'esordio inserito è
iniziata

```

```

    WHERE DATE_ADD(T.DataInizio, INTERVAL T.Durata DAY) = (SELECT
DataInizio

```

```

        FROM Terapia

```

```

        WHERE Codice = NEW.Terapia)

```

```

        AND E.Patologia = NEW.Patologia

```

```

        AND E.Animale = NEW.Animale;

```

```

DECLARE CONTINUE HANDLER FOR NOT FOUND
SET finito = 1;

-- Se l'animale non è già in quarantena
IF (SELECT Quarantena FROM Animale WHERE Codice = NEW.Animale) = 0
THEN

    OPEN cur;
    scan: LOOP
        FETCH cur INTO data_inizio;
        IF finito = 1 THEN
            LEAVE scan;
        END IF;

        -- Controllo se c'è una terapia, per lo stesso animale e la
        stessa patologia,
        -- che si è conclusa il giorno in cui è iniziata la terapia
        che sto scorrendo
        -- (in altre parole: la terapia precedente non ha avuto
        successo)

        SELECT COUNT(*) INTO terapie_fallite
        FROM Terapia T
        INNER JOIN
        Esordio E ON E.Terapia = T.Codice -- Terapie prescritte a
        seguito di una visita
        WHERE E.Animale = NEW.Animale
        AND E.Patologia = NEW.Patologia
        AND DATE_ADD(T.DataInizio, INTERVAL T.Durata DAY) =
        data_inizio;

        -- Metto l'animale in quarantena
        IF terapie_fallite > 0 THEN
            UPDATE Animale
            SET Quarantena = 1
            WHERE Codice = NEW.Animale;
        END IF;
    END LOOP;

    CLOSE cur;
END IF;
END $$
DELIMITER ;

-- -----
--      Popolamento      --
-- -----
INSERT INTO `Agriturismo` (Indirizzo) VALUES
('Via Montanara 7, Greve (FI)'),
('Via IV Novembre 9, Ariccia (RM)'),
('Via C.Cassola 104, Rivoli (TO)'),
('Via Umbria 16, Bibbona (LI)'),
('Via S. Marino 51, Teggiano (SA)'),
('Via Tripoli 32, Verucchio (RN)');

INSERT INTO `Stalla` (Zona, Agriturismo) VALUES
('A', 1),
('B', 1),
('A', 2),

```

```

('A', 3),
('A', 4),
('A', 5),
('A', 6),
('B', 6);

```

```

INSERT INTO `Specie` VALUES
('Bovini', 'Bovinae'),
('Ovini', 'Aegodontia'),
('Caprini', 'Caprinae');

```

```

INSERT INTO `Locale` (Pavimentazione, Lunghezza, Larghezza, Altezza, Stalla,
Specie, NumeroMassimoAnimali) VALUES
('Calcestruzzo', 70, 70, 2.5, 1, 'Bovini', 20),
('Calcestruzzo', 50, 50, 2.5, 2, 'Bovini', 15),
('Calcestruzzo', 50, 50, 2.5, 2, 'Bovini', 15),
('Calcestruzzo', 55, 55, 2.5, 3, 'Bovini', 16),
('Calcestruzzo', 20, 20, 2, 4, 'Ovini', 8),
('Calcestruzzo', 65, 65, 2, 4, 'Caprini', 20),
('Calcestruzzo', 30, 30, 2.5, 5, 'Bovini', 8),
('Calcestruzzo', 50, 50, 2, 6, 'Ovini', 25),
('Calcestruzzo', 45, 45, 2.5, 7, 'Bovini', 12),
('Calcestruzzo', 50, 50, 2, 8, 'Caprini', 14);

```

```

INSERT INTO `StatoAria` VALUES
('2019-09-21 15:00:07', 1, 18, 70),
('2019-09-20 17:49:32', 2, 16, 80),
('2019-09-24 15:06:23', 3, 14, 75),
('2019-09-22 10:34:12', 4, 13, 76),
('2019-09-19 21:33:02', 5, 15, 62),
('2019-09-23 13:04:39', 6, 15, 67),
('2019-09-20 17:49:32', 7, 14, 82),
('2019-09-20 16:18:29', 8, 13, 71),
('2019-09-21 20:12:00', 9, 16, 64),
('2019-09-25 09:08:45', 10, 13, 79);

```

```

INSERT INTO `InterventoPulizia` VALUES
('2019-09-22 10:34:12', 1, 'Effettuato', 65, 45, 30),
('2019-09-21 12:56:39', 2, 'Effettuato', 63, 60, 21),
('2019-09-20 09:12:45', 3, 'Effettuato', 47, 52, 45),
('2019-09-22 14:44:10', 4, 'Effettuato', 48, 60, 41),
('2019-09-22 03:32:01', 5, 'Effettuato', 56, 60, 15),
('2019-09-23 10:28:51', 6, 'Effettuato', 65, 59, 40),
('2019-09-19 08:33:49', 7, 'Effettuato', 30, 10, 40),
('2019-09-19 11:00:37', 8, 'Effettuato', 64, 23, 45),
('2019-09-16 01:34:12', 9, 'Effettuato', 34, 60, 44),
('2019-09-24 23:45:58', 9, 'Richiesto', 51, 60, 40),
('2019-09-20 12:25:22', 10, 'Effettuato', 65, 39, 26),
('2019-09-25 04:12:09', 10, 'Richiesto', 65, 48, 39);

```

```

INSERT INTO `Abbeveratoio` (Capienza, Locale) VALUES
(80, 1),
(60, 2), (60, 2),
(60, 3), (100, 3), (120, 3),
(110, 4), (180, 4),
(200, 5),
(180, 6),
(80, 7), (60, 7), (60, 7),
(120, 8), (60, 8),
(180, 9),

```

```
(120, 10), (120, 10);
```

```
INSERT INTO `StatoAbbeveratoio` VALUES
('2019-09-21 15:00:07', 1, 77), ('2019-09-21 16:13:39', 1, 64), ('2019-09-23
08:31:51', 1, 80),
('2019-09-23 17:14:50', 2, 53), ('2019-09-24 19:55:00', 2, 39),
('2019-09-24 23:00:00', 3, 60),
('2019-09-20 13:15:19', 4, 56),
('2019-09-20 14:43:23', 5, 96),
('2019-09-25 07:56:33', 6, 109),
('2019-09-25 08:11:40', 7, 60),
('2019-09-24 09:49:42', 8, 163),
('2019-09-22 11:10:09', 9, 171),
('2019-09-22 12:34:45', 10, 99), ('2019-09-22 18:54:17', 10, 61),
('2019-09-23 13:04:39', 11, 69),
('2019-09-20 17:49:32', 12, 43),
('2019-09-20 16:18:29', 13, 60),
('2019-09-21 20:12:00', 14, 101),
('2019-09-19 21:33:02', 15, 49),
('2019-09-19 10:45:46', 16, 180),
('2019-09-25 08:36:32', 17, 89),
('2019-09-24 15:06:23', 18, 117), ('2019-09-24 22:37:21', 18, 81);
```

```
INSERT INTO `SostanzaChimica` VALUES
('Roslen', 80, 40),
('Tombow', 60, 20),
('Pritted', 60, 60);
```

```
INSERT INTO `ProprietaAcqua` VALUES
('2019-09-21 15:00:07', 1, 'Roslen'),
('2019-09-20 14:43:23', 5, 'Roslen'),
('2019-09-25 08:11:40', 7, 'Roslen'),
('2019-09-22 18:54:17', 10, 'Pritted'),
('2019-09-20 17:49:32', 12, 'Tombow'),
('2019-09-19 21:33:02', 15, 'Tombow'),
('2019-09-24 15:06:23', 18, 'Pritted'),
('2019-09-24 22:37:21', 18, 'Pritted');
```

```
INSERT INTO `Razza` VALUES
('Frisona', 'Bovini'),
('Chianina', 'Bovini'),
('Simmental', 'Bovini'),
('Sarda', 'Ovini'),
('Capra boera', 'Caprini');
```

```
INSERT INTO `Fornitore` VALUES
('01062556495', 'Marchini & Co. S.n.c', 'Francesca', 'Via del grappolo 7,
Greve (FI)'),
('01162783496', 'Happy Animals S.s', 'Andrea', 'Via degli olivi 39, Ariccia
(RM)'),
('01345556495', 'Da filippo S.s', 'Filippo', 'Via della mappa 58, Torino
(TO)'),
('02132332495', 'Verdi & Rossi S.a.p.a', 'Marino', 'Via Marco Polo 43,
Teggiano (SA)'),
('01062559533', 'L'allevamento per te S.s', 'Giorgio', 'Via Podgara 4,
Livorno (LI)'),
('10362543283', 'Guerrini S.r.l', 'Christian', 'Via della Rimembranza 109,
Verucchio (RN)');
```

```
INSERT INTO `Veterinario` (Nome, Cognome, Telefono, Agriturismo) VALUES
```

```

('Marco', 'Ciano', '3381279385', 1),
('Amerigo', 'Azzurro', '3661938293', 2),
('Luca', 'Celesti', '3335830395', 2),
('Gina', 'Rosei', '3336132111', 3),
('Martina', 'Grigi', '3354850840', 4),
('Giuseppe', 'Rossini', '3395753391', 5),
('Lucia', 'Bianchetti', '3314849321', 6);

-- Animali acquistati
INSERT INTO `Animale` (Peso, Sesso, Razza, DataNascita, Altezza, Tipo,
Locale) VALUES
(956, 'M', 'Frisona', '2015-07-05', 138, 'Acquistato', 2),
(700, 'F', 'Frisona', '2014-05-01', 142, 'Acquistato', 1),
(655, 'F', 'Frisona', '2016-03-12', 143, 'Acquistato', 2),
(1080, 'M', 'Chianina', '2013-10-29', 152, 'Acquistato', 3),
(717, 'F', 'Chianina', '2011-12-30', 113, 'Acquistato', 4),
(901, 'M', 'Chianina', '2010-03-27', 129, 'Acquistato', 4),
(600, 'M', 'Simmental', '2018-01-03', 101, 'Acquistato', 7),
(978, 'M', 'Simmental', '2009-06-15', 150, 'Acquistato', 7),
(881, 'M', 'Chianina', '2009-08-12', 137, 'Acquistato', 9),
(784, 'F', 'Chianina', '2015-08-09', 122, 'Acquistato', 9),

(151, 'M', 'Sarda', '2015-10-10', 109, 'Acquistato', 5),
(145, 'F', 'Sarda', '2014-01-05', 108, 'Acquistato', 5),
(98, 'M', 'Sarda', '2009-06-23', 90, 'Acquistato', 8),
(85, 'F', 'Sarda', '2009-07-19', 130, 'Acquistato', 8),

(110, 'M', 'Capra boera', '2014-04-12', 45, 'Acquistato', 6),
(131, 'F', 'Capra boera', '2015-07-18', 60, 'Acquistato', 6),
(109, 'M', 'Capra boera', '2016-09-04', 54, 'Acquistato', 10),
(120, 'F', 'Capra boera', '2016-08-14', 49, 'Acquistato', 10);

INSERT INTO `Acquisto` (Animale, Fornitore, DataArrivo, DataAcquisto) VALUES
(1, '01062556495', '2016-08-12', '2016-08-01'),
(2, '01062556495', '2015-10-03', '2015-09-29'),
(3, '01062556495', '2016-10-10', '2016-10-01'),
(4, '01062556495', '2015-08-19', '2015-08-17'),
(5, '01162783496', '2013-07-29', '2013-07-19'),
(6, '01162783496', '2014-05-01', '2014-04-20'),
(7, '01062559533', '2016-12-19', '2016-12-08'),
(8, '01062559533', '2010-12-16', '2010-12-10'),
(9, '10362543283', '2012-01-12', '2011-12-28'),
(10, '10362543283', '2018-03-09', '2018-02-25'),
(11, '01345556495', '2016-01-09', '2016-01-01'),
(12, '01345556495', '2014-08-10', '2014-08-02'),
(13, '02132332495', '2012-07-07', '2012-06-28'),
(14, '02132332495', '2012-06-19', '2012-06-07'),
(15, '01345556495', '2014-11-30', '2014-11-20'),
(16, '01345556495', '2015-10-29', '2015-10-20'),
(17, '10362543283', '2018-01-05', '2017-12-20'),
(18, '10362543283', '2018-05-04', '2018-04-28');

INSERT INTO `Riproduzione` VALUES
('2018-04-09 12:29:10', 1, 2, 'Successo', 1),
('2014-09-03 15:01:39', 5, 6, 'Successo', 1),
('2019-01-25 13:19:00', 13, 14, 'Successo', 6),
('2015-11-12 13:59:28', 15, 16, 'Successo', 4),
('2018-07-09 14:40:40', 17, 18, 'Successo', 7),
('2017-03-23 12:51:26', 11, 12, 'Insuccesso', 4);

```

```
INSERT INTO `Gestazione` (Responsabile, Istante, Padre, Madre) VALUES
(1, '2018-04-09 12:29:10', 1, 2),
(1, '2014-09-03 15:01:39', 5, 6),
(5, '2019-01-25 13:19:00', 13, 14),
(3, '2015-11-12 13:59:28', 15, 16),
(6, '2018-07-09 14:40:40', 17, 18);
```

```
INSERT INTO `Complicanza` VALUES
(5, '2018-11-13', 'La gestante ha sviluppato la Brucellosi, rilevata a seguito di una Visita.');
```

```
INSERT INTO `Controllo` (Stato, DataProgrammata, DataEffettiva, Esito, Gestazione, Veterinario) VALUES
('Svolto', '2018-05-08', '2018-05-08', 'Negativo', 1, 1),
('Svolto', '2018-06-08', '2018-06-08', 'Positivo', 1, 1),
('Svolto', '2014-10-03', '2014-10-05', 'Positivo', 2, 1),
('Svolto', '2019-03-01', '2019-03-02', 'Positivo', 3, 5),
('Svolto', '2015-12-12', '2015-12-12', 'Positivo', 4, 3),
('Svolto', '2018-08-15', '2018-08-15', 'Positivo', 5, 6);
```

```
INSERT INTO `Esame` (Nome, `Data`, Macchinario, Descrizione, CodiceControllo) VALUES
('Verifica cardiologica', '2018-05-12', NULL, 'Check-up cardiaco per escludere la presenza di problemi cardiaci', 1),
('Verifica conformità muscolo cardiaco', '2018-05-12', 'Ecografo', 'Controllo della corretta formazione del cuore.', 1);
```

```
INSERT INTO `Patologia` VALUES
('Afta epizootica', 'Bocca'),
('Colibacillosi', 'Utero'),
('Febbre Q', 'Generale'),
('Paratubercolosi', 'Intestino'),
('Rinotracheite bovina', 'Polmoni'),
('Gastroenterotossiemia', 'Stomaco'),
('Brucellosi', 'Generale');
```

```
INSERT INTO `Farmaco` VALUES
('Afta med scudo', 2, 'Acido ialuronico'),
('Ciproxin', 3, 'Ciprofloxacina'),
('Normix', 4, 'Rifaximina'),
('Cotrimossazolo', 2, 'Sulfametossazolo'),
('Rifadin', 1, 'Rifampicina');
```

```
INSERT INTO `OrarioSomministrazione` VALUES
('10:00:00', 'Afta med scudo'),
('20:00:00', 'Afta med scudo'),
('12:00:00', 'Ciproxin'),
('21:00:00', 'Ciproxin'),
('06:00:00', 'Ciproxin'),
('06:00:00', 'Normix'),
('12:00:00', 'Normix'),
('18:00:00', 'Normix'),
('22:00:00', 'Normix'),
('10:00:00', 'Cotrimossazolo'),
('21:00:00', 'Cotrimossazolo'),
('15:00:00', 'Rifadin');
```

-- Animali nati

```
INSERT INTO `Animale` (Peso, Sesso, Razza, DataNascita, Altezza, Tipo, Locale) VALUES
```

```
(980, 'M', 'Frisona', '2019-01-17', 150, 'Nato', 1),
(680, 'F', 'Chianina', '2015-07-10', 131, 'Nato', 2),
(570, 'F', 'Chianina', '2015-07-10', 109, 'Nato', 3),

(61, 'F', 'Sarda', '2019-06-01', 95, 'Nato', 8),

(117, 'F', 'Capra boera', '2016-02-21', 58, 'Nato', 6);
```

```
INSERT INTO `Nascita` (Animale, Gestazione) VALUES
(19, 1),
(20, 2),
(21, 2),
(22, 3),
(23, 4);
```

```
INSERT INTO `ZonaPascolo` VALUES
('A1', 1),
('B7', 1),
('D', 2),
('A1', 3),
('C1', 4),
('E', 5),
('A1', 6),
('A2', 6);
```

```
INSERT INTO `Recinzione` (LatitudineIniziale, LongitudineIniziale,
LatitudineFinale, LongitudineFinale, Tipo, ZonaPascolo, Agriturismo) VALUES
(43.59185, 11.31299, 43.59402, 11.31004, 'Fissa', 'A1', 1),
(43.59185, 11.31299, 43.59293, 11.31004, 'Mobile', 'A1', 1),
(43.59293, 11.31299, 43.59402, 11.31004, 'Mobile', 'A1', 1),
(43.59868, 11.30172, 43.59670, 11.30504, 'Fissa', 'B7', 1),
(43.59868, 11.30172, 43.59769, 11.30504, 'Mobile', 'B7', 1),
(43.59769, 11.30172, 43.59670, 11.30504, 'Mobile', 'B7', 1),
(41.71944, 12.67385, 41.71749, 12.67688, 'Fissa', 'D', 2),
(41.71944, 12.67385, 41.71846, 12.67688, 'Mobile', 'D', 2),
(41.71847, 12.67385, 41.71749, 12.67688, 'Mobile', 'D', 2),
(45.06198, 7.49848, 45.05934, 7.50071, 'Fissa', 'A1', 3),
(45.06198, 7.49848, 45.06066, 7.50071, 'Mobile', 'A1', 3),
(45.06066, 7.49848, 45.05934, 7.50071, 'Mobile', 'A1', 3),
(43.26912, 10.60461, 43.26749, 10.60101, 'Fissa', 'C1', 4),
(43.26912, 10.60461, 43.26830, 10.60101, 'Mobile', 'C1', 4),
(43.26831, 10.60461, 43.26749, 10.60101, 'Mobile', 'C1', 4),
(40.37298, 15.54324, 40.37032, 15.54482, 'Fissa', 'E', 5),
(40.37298, 15.54324, 40.37165, 15.54482, 'Mobile', 'E', 5),
(40.37165, 15.54324, 40.37032, 15.54482, 'Mobile', 'E', 5),
(43.98036, 12.42323, 43.97843, 12.41969, 'Fissa', 'A1', 6),
(43.98036, 12.42323, 43.97939, 12.41969, 'Mobile', 'A1', 6),
(43.97940, 12.42323, 43.97843, 12.41969, 'Mobile', 'A1', 6),
(43.97843, 12.41969, 43.97682, 12.42305, 'Fissa', 'A2', 6),
(43.97843, 12.41969, 43.97762, 12.42305, 'Mobile', 'A2', 6),
(43.97763, 12.41969, 43.97682, 12.42305, 'Mobile', 'A2', 6);
```

```
INSERT INTO `Posizione` VALUES
('2019-09-24 16:55:36', 1, 43.59246, 11.31209), -- Recinzione: 2
('2019-09-18 14:25:42', 1, 43.59246, 11.31209),
('2019-09-20 16:50:38', 1, 43.59246, 11.31209),
('2019-09-19 14:01:42', 1, 43.59246, 11.31209),
('2019-09-20 14:03:45', 1, 43.59246, 11.31209),
('2019-09-20 08:49:17', 1, 43.59246, 11.31209),
('2019-09-18 16:08:22', 1, 43.59246, 11.31209),
```

```

('2019-09-19 17:20:35', 1, 43.59246, 11.31209),
('2019-09-20 09:17:40', 1, 43.59246, 11.31209),
('2019-09-24 13:39:51', 1, 43.59246, 11.31209),
('2019-09-24 10:30:56', 2, 43.59378, 11.31238), -- Recinzione: 3
('2019-09-19 10:22:53', 2, 43.59378, 11.31238),
('2019-09-19 08:12:01', 2, 43.59378, 11.31238),
('2019-09-20 14:15:33', 2, 43.59324, 11.31142),
('2019-09-22 16:41:34', 2, 43.59324, 11.31142),
('2019-09-22 16:47:43', 2, 43.59324, 11.31142),
('2019-09-22 10:13:47', 2, 43.59359, 11.31028),
('2019-09-23 10:45:41', 2, 43.59381, 11.31103),
('2019-09-24 09:33:36', 2, 43.59381, 11.31103),
('2019-09-22 08:30:51', 2, 43.59381, 11.31103),
('2019-09-19 10:13:20', 3, 43.59243, 11.31205), -- Recinzione: 2
('2019-09-24 14:40:39', 3, 43.59243, 11.31205),
('2019-09-22 15:03:10', 3, 43.59243, 11.31205),
('2019-09-18 07:05:57', 3, 43.59243, 11.31205),
('2019-09-21 14:08:18', 3, 43.59243, 11.31205),
('2019-09-18 10:50:12', 3, 43.59243, 11.31205),
('2019-09-19 13:52:36', 3, 43.59243, 11.31205),
('2019-09-23 16:51:34', 3, 43.59243, 11.31205),
('2019-09-24 15:06:34', 3, 43.59243, 11.31205),
('2019-09-21 17:04:45', 3, 43.59243, 11.31205),
('2019-09-18 09:02:25', 4, 43.59724, 11.30414), -- Recinzione: 6
('2019-09-18 07:26:10', 4, 43.59724, 11.30414),
('2019-09-19 17:03:30', 4, 43.59724, 11.30414),
('2019-09-21 17:12:10', 4, 43.59675, 11.30464),
('2019-09-21 16:02:42', 4, 43.59675, 11.30464),
('2019-09-18 15:42:34', 4, 43.59675, 11.30464),
('2019-09-19 10:31:18', 4, 43.59750, 11.30202),
('2019-09-20 17:25:19', 4, 43.59760, 11.30224),
('2019-09-21 13:42:38', 4, 43.59768, 11.30401),
('2019-09-21 15:53:45', 4, 43.59768, 11.30401),
('2019-09-19 16:21:01', 5, 41.71870, 12.67569), -- Recinzione: 8 (GIORNO
DIVERSO DALL'ANIMALE 6!)
('2019-09-18 16:04:34', 5, 41.71870, 12.67569),
('2019-09-18 13:39:41', 5, 41.71893, 12.67476),
('2019-09-19 07:17:54', 5, 41.71893, 12.67476),
('2019-09-20 08:23:41', 5, 41.71893, 12.67476),
('2019-09-19 07:17:58', 5, 41.71895, 12.67522),
('2019-09-20 17:25:19', 5, 41.71895, 12.67528),
('2019-09-18 16:25:56', 5, 41.71895, 12.67522),
('2019-09-18 14:53:34', 5, 41.71848, 12.67642),
('2019-09-19 07:01:57', 5, 41.71893, 12.67640),
('2019-09-21 07:07:24', 6, 41.71777, 12.67564), -- Recinzione: 9 (GIORNO
DIVERSO DALL'ANIMALE 5!)
('2019-09-22 09:47:48', 6, 41.71777, 12.67564),
('2019-09-22 16:56:41', 6, 41.71777, 12.67564),
('2019-09-23 09:26:37', 6, 41.71777, 12.67564),
('2019-09-22 16:17:17', 6, 41.71756, 12.67526),
('2019-09-24 13:49:17', 6, 41.71823, 12.67517),
('2019-09-21 14:21:10', 6, 41.71823, 12.67517),
('2019-09-24 17:24:33', 6, 41.71823, 12.67517),
('2019-09-22 15:19:06', 6, 41.71823, 12.67517),
('2019-09-24 08:35:07', 6, 41.71765, 12.67472),
('2019-09-20 09:07:53', 7, 43.26906, 10.60147), -- Recinzione: 14
('2019-09-18 08:20:13', 7, 43.26906, 10.60147),
('2019-09-20 14:19:08', 7, 43.26906, 10.60147),
('2019-09-19 14:20:40', 7, 43.26906, 10.60147),
('2019-09-19 07:59:01', 7, 43.26877, 10.60449),

```



```

('2019-09-18 08:12:55', 7, 43.26877, 10.60449),
('2019-09-18 10:07:28', 7, 43.26896, 10.60360),
('2019-09-19 07:18:30', 7, 43.26896, 10.60360),
('2019-09-20 17:09:57', 7, 43.26896, 10.60360),
('2019-09-20 09:20:48', 7, 43.26874, 10.60454),
('2019-09-21 08:26:24', 8, 43.26757, 10.60321), -- Recinzione: 15
('2019-09-24 08:04:26', 8, 43.26757, 10.60321),
('2019-09-22 16:47:28', 8, 43.26757, 10.60321),
('2019-09-23 15:55:23', 8, 43.26757, 10.60321),
('2019-09-24 14:52:27', 8, 43.26820, 10.60440),
('2019-09-21 08:58:56', 8, 43.26820, 10.60440),
('2019-09-21 08:03:52', 8, 43.26820, 10.60440),
('2019-09-23 07:49:07', 8, 43.26765, 10.60393),
('2019-09-22 16:53:53', 8, 43.26772, 10.60274),
('2019-09-24 17:08:36', 8, 43.26772, 10.60274),
('2019-09-18 14:28:07', 9, 43.98019, 12.42237), -- Recinzione: 20
('2019-09-18 17:29:10', 9, 43.97942, 12.42144),
('2019-09-20 17:27:17', 9, 43.98024, 12.41987),
('2019-09-19 08:13:36', 9, 43.98024, 12.41987),
('2019-09-18 07:17:13', 9, 43.98024, 12.41987),
('2019-09-19 07:58:30', 9, 43.98024, 12.41987),
('2019-09-18 09:00:45', 9, 43.98024, 12.41987),
('2019-09-20 15:21:28', 9, 43.98024, 12.41987),
('2019-09-20 09:58:17', 9, 43.97976, 12.42089),
('2019-09-20 15:00:16', 9, 43.97976, 12.42089),
('2019-09-24 08:51:48', 10, 43.97879, 12.42104), -- Recinzione: 21
('2019-09-22 15:32:17', 10, 43.97879, 12.42104),
('2019-09-21 07:39:31', 10, 43.97874, 12.42090),
('2019-09-22 10:47:19', 10, 43.97874, 12.42090),
('2019-09-21 10:11:17', 10, 43.97874, 12.42090),
('2019-09-23 13:41:02', 10, 43.97874, 12.42090),
('2019-09-23 10:57:51', 10, 43.97905, 12.41985),
('2019-09-23 16:59:24', 10, 43.97905, 12.41985),
('2019-09-21 07:23:19', 10, 43.97905, 12.41985),
('2019-09-22 07:04:58', 10, 43.97896, 12.42231),
('2019-09-18 08:50:31', 11, 45.06080, 7.50014), -- Recinzione: 11
('2019-09-20 17:05:32', 11, 45.06080, 7.50014),
('2019-09-18 07:46:10', 11, 45.06080, 7.50014),
('2019-09-23 09:27:59', 11, 45.06189, 7.50027),
('2019-09-20 07:07:39', 11, 45.06189, 7.50027),
('2019-09-21 14:26:58', 11, 45.06189, 7.50027),
('2019-09-18 16:51:26', 11, 45.06189, 7.50027),
('2019-09-19 10:32:44', 11, 45.06186, 7.49901),
('2019-09-19 07:56:41', 11, 45.06122, 7.50070),
('2019-09-18 08:38:20', 11, 45.06104, 7.50050),
('2019-09-18 08:50:22', 12, 45.06159, 7.49896), -- Recinzione: 11
('2019-09-20 17:05:52', 12, 45.06123, 7.50024),
('2019-09-18 07:46:38', 12, 45.06123, 7.50024),
('2019-09-23 09:28:11', 12, 45.06123, 7.50024),
('2019-09-20 07:07:18', 12, 45.06121, 7.49938),
('2019-09-21 14:26:29', 12, 45.06121, 7.49938),
('2019-09-18 16:51:25', 12, 45.06121, 7.49938),
('2019-09-19 10:33:00', 12, 45.06121, 7.49938),
('2019-09-19 07:56:41', 12, 45.06121, 7.49938),
('2019-09-18 08:38:40', 12, 45.06151, 7.49894),
('2019-09-21 14:40:36', 13, 40.37264, 15.54466), -- Recinzione: 17
('2019-09-22 08:29:58', 13, 40.37264, 15.54466),
('2019-09-23 09:27:58', 13, 40.37264, 15.54466),
('2019-09-24 08:49:48', 13, 40.37179, 15.54437),
('2019-09-21 10:18:19', 13, 40.37228, 15.54474),

```

```

('2019-09-22 10:55:24', 13, 40.37228, 15.54474),
('2019-09-24 14:03:30', 13, 40.37228, 15.54474),
('2019-09-24 10:31:30', 13, 40.37239, 15.54375),
('2019-09-22 16:59:43', 13, 40.37239, 15.54375),
('2019-09-21 15:04:35', 13, 40.37239, 15.54375),
('2019-09-20 10:01:21', 14, 40.37065, 15.54342), -- Recinzione: 18
('2019-09-19 10:13:36', 14, 40.37065, 15.54342),
('2019-09-18 13:45:23', 14, 40.37065, 15.54342),
('2019-09-18 14:28:23', 14, 40.37065, 15.54412),
('2019-09-20 10:18:58', 14, 40.37065, 15.54412),
('2019-09-20 09:16:14', 14, 40.37075, 15.54446),
('2019-09-18 16:44:10', 14, 40.37075, 15.54446),
('2019-09-19 08:58:44', 14, 40.37075, 15.54446),
('2019-09-19 16:47:40', 14, 40.37161, 15.54459),
('2019-09-18 17:22:05', 14, 40.37161, 15.54459),
('2019-09-20 10:57:57', 15, 45.05962, 7.49991), -- Recinzione: 12
('2019-09-19 08:38:39', 15, 45.05962, 7.49991),
('2019-09-21 15:27:15', 15, 45.05962, 7.49991),
('2019-09-24 16:19:30', 15, 45.05962, 7.49991),
('2019-09-18 13:45:37', 15, 45.05962, 7.49991),
('2019-09-20 08:25:39', 15, 45.05962, 7.49991),
('2019-09-20 08:38:18', 15, 45.06064, 7.49908),
('2019-09-18 14:07:43', 15, 45.06064, 7.49908),
('2019-09-21 07:50:31', 15, 45.06064, 7.49908),
('2019-09-22 08:08:29', 15, 45.06064, 7.49908),
('2019-09-20 10:57:43', 16, 45.05964, 7.49991), -- Recinzione: 12
('2019-09-19 08:38:15', 16, 45.05964, 7.49991),
('2019-09-21 15:27:00', 16, 45.05964, 7.49991),
('2019-09-24 16:19:59', 16, 45.05964, 7.49991),
('2019-09-18 13:45:10', 16, 45.05964, 7.49991),
('2019-09-20 08:25:38', 16, 45.05964, 7.49991),
('2019-09-20 08:38:25', 16, 45.05934, 7.50018),
('2019-09-18 14:07:44', 16, 45.05934, 7.50018),
('2019-09-21 07:50:41', 16, 45.05934, 7.50018),
('2019-09-22 08:08:38', 16, 45.05934, 7.50018),
('2019-09-19 13:51:08', 17, 43.97840, 12.42039), -- Recinzione: 23
('2019-09-20 14:14:49', 17, 43.97840, 12.42039),
('2019-09-19 17:07:16', 17, 43.97840, 12.42039),
('2019-09-20 13:43:46', 17, 43.97765, 12.42301),
('2019-09-18 13:56:37', 17, 43.97765, 12.42301),
('2019-09-20 15:25:03', 17, 43.97765, 12.42301),
('2019-09-19 10:21:58', 17, 43.97822, 12.42196),
('2019-09-19 16:53:21', 17, 43.97787, 12.42097),
('2019-09-18 08:49:44', 17, 43.97787, 12.42097),
('2019-09-20 09:41:49', 17, 43.97787, 12.42097),
('2019-09-21 16:57:35', 18, 43.97694, 12.42014), -- Recinzione: 24
('2019-09-23 09:11:51', 18, 43.97694, 12.42014),
('2019-09-23 17:21:40', 18, 43.97694, 12.42014),
('2019-09-21 16:38:18', 18, 43.97694, 12.42014),
('2019-09-21 09:13:19', 18, 43.97694, 12.42014),
('2019-09-22 08:24:57', 18, 43.97694, 12.42014),
('2019-09-23 09:31:20', 18, 43.97682, 12.42260),
('2019-09-24 10:41:42', 18, 43.97695, 12.42223),
('2019-09-24 10:47:45', 18, 43.97748, 12.42006),
('2019-09-23 07:00:14', 18, 43.97748, 12.42006),
('2019-09-20 14:25:33', 19, 43.59370, 11.31235), -- Recinzione: 3
('2019-09-24 14:24:13', 19, 43.59376, 11.31240),
('2019-09-19 16:39:06', 19, 43.59376, 11.31240),
('2019-09-18 14:02:41', 19, 43.59320, 11.31142),
('2019-09-19 10:16:04', 19, 43.59320, 11.31142),

```

```

('2019-09-18 10:32:08', 19, 43.59320, 11.31142),
('2019-09-18 10:52:21', 19, 43.59358, 11.31025),
('2019-09-20 08:02:28', 19, 43.59380, 11.31100),
('2019-09-21 13:49:47', 19, 43.59380, 11.31100),
('2019-09-19 14:14:45', 19, 43.59380, 11.31100),
('2019-09-24 15:17:25', 20, 43.59263, 11.31163), -- Recinzione: 2
('2019-09-21 09:25:29', 20, 43.59263, 11.31163),
('2019-09-20 15:21:42', 20, 43.59263, 11.31163),
('2019-09-21 16:34:34', 20, 43.59263, 11.31163),
('2019-09-22 14:30:13', 20, 43.59278, 11.31286),
('2019-09-23 09:53:54', 20, 43.59278, 11.31286),
('2019-09-23 17:12:04', 20, 43.59278, 11.31286),
('2019-09-24 14:25:14', 20, 43.59278, 11.31286),
('2019-09-22 08:36:30', 20, 43.59219, 11.31186),
('2019-09-19 14:10:28', 20, 43.59219, 11.31186),
('2019-09-18 10:21:18', 21, 43.59720, 11.30412), -- Recinzione: 6
('2019-09-19 15:05:27', 21, 43.59720, 11.30412),
('2019-09-18 14:06:36', 21, 43.59720, 11.30412),
('2019-09-24 10:43:05', 21, 43.59670, 11.30463),
('2019-09-22 09:54:40', 21, 43.59670, 11.30463),
('2019-09-22 09:06:51', 21, 43.59670, 11.30463),
('2019-09-23 16:56:48', 21, 43.59755, 11.30202),
('2019-09-22 07:49:59', 21, 43.59755, 11.30202),
('2019-09-18 16:32:47', 21, 43.59767, 11.30399),
('2019-09-20 14:07:11', 21, 43.59767, 11.30399),
('2019-09-22 14:44:24', 22, 40.37291, 15.54400), -- Recinzione: 17
('2019-09-24 16:04:26', 22, 40.37291, 15.54400),
('2019-09-23 09:22:09', 22, 40.37291, 15.54400),
('2019-09-21 10:29:17', 22, 40.37196, 15.54347),
('2019-09-21 14:18:42', 22, 40.37186, 15.54471),
('2019-09-22 17:12:16', 22, 40.37186, 15.54471),
('2019-09-21 09:31:46', 22, 40.37186, 15.54471),
('2019-09-22 08:01:54', 22, 40.37208, 15.54328),
('2019-09-23 15:54:07', 22, 40.37208, 15.54328),
('2019-09-24 09:38:05', 22, 40.37208, 15.54328),
('2019-09-20 10:57:44', 23, 45.06010, 7.49949), -- Recinzione: 12
('2019-09-19 08:38:15', 23, 45.06010, 7.49949),
('2019-09-21 15:27:01', 23, 45.06010, 7.49949),
('2019-09-24 16:20:10', 23, 45.06010, 7.49949),
('2019-09-18 13:45:12', 23, 45.06048, 7.50041),
('2019-09-20 08:25:41', 23, 45.06048, 7.50041),
('2019-09-20 08:38:05', 23, 45.06048, 7.50041),
('2019-09-18 14:07:24', 23, 45.06048, 7.50041),
('2019-09-21 07:50:49', 23, 45.06048, 7.50041),
('2019-09-22 08:08:29', 23, 45.06048, 7.50041);

```

```

INSERT INTO `Visita` VALUES

```

```

('2019-09-18 16:10:28', 1, 1), ('2019-09-24 09:52:56', 1, 1),
('2019-07-19 17:08:31', 2, 1),
('2018-12-28 12:53:58', 3, 1),
('2017-03-13 13:23:01', 4, 1),
('2019-09-18 17:00:59', 5, 2),
('2019-08-30 09:01:08', 6, 3), ('2019-09-09 08:54:38', 6, 3), ('2019-09-19
10:12:06', 6, 3),
('2019-04-19 09:10:38', 7, 5),
('2018-05-22 08:31:33', 8, 5),
('2019-04-25 12:19:52', 9, 7),
('2019-04-25 15:40:40', 10, 7),
('2019-04-28 15:27:29', 11, 4),
('2019-05-04 11:40:56', 12, 4),

```

```

('2019-07-20 10:04:00', 13, 6),
('2019-08-29 16:32:21', 14, 6), ('2019-09-23 13:14:09', 14, 6),
('2019-02-03 11:50:59', 15, 4),
('2018-09-28 16:58:08', 16, 4),
('2018-08-01 14:23:02', 17, 7),
('2018-11-13 16:10:28', 18, 7), ('2018-12-01 15:16:01', 18, 7),
('2019-01-17 11:08:17', 19, 1),
('2015-07-10 12:34:39', 20, 1),
('2015-07-10 12:45:57', 21, 1),
('2019-06-01 18:56:01', 22, 6),
('2016-02-21 16:28:14', 23, 4);

```

```

INSERT INTO `IndicatoreBenessere` VALUES
('Livello deambulazione'), -- Ottimo, Medio, Basso o Assente
('Livello idratazione'), -- Da 0 a 100%
('Emocromo'),
('Funzionalità fegato'), -- Ok o Rilevata disfunzione
('Funzionalità cuore'), -- Ok o Rilevata disfunzione
('Funzionalità pancreas'), -- Ok o Rilevata disfunzione
('Risposta oculare'), -- Elevata, Basso o Assente
('Lucentezza pelo'), -- Lucente, A chiazze, Spento
('Respirazione'), -- Corretta o Affannata
('Spessore zoccolo'), -- Livello da 0 a 10
('Massa magra'), -- Da 0 a 100%
('Massa grassa'), -- Da 0 a 100%
('Livello vigilanza'); -- Da 0 a 5

```

```

INSERT INTO `Monitoraggio` VALUES
('2019-09-18 16:10:28', 1, 'Livello deambulazione', 'Ottimo'), ('2019-09-18
16:10:28', 1, 'Livello idratazione', '80%'), ('2019-09-18 16:10:28', 1,
'Risposta oculare', 'Elevata'), ('2019-09-18 16:10:28', 1, 'Spessore
zoccolo', 10),
('2019-07-19 17:08:31', 2, 'Funzionalità fegato', 'Ok'), ('2019-07-19
17:08:31', 2, 'Funzionalità cuore', 'Ok'), ('2019-07-19 17:08:31', 2,
'Funzionalità pancreas', 'Ok'), ('2019-07-19 17:08:31', 2, 'Livello
vigilanza', 4),
('2019-08-30 09:01:08', 6, 'Massa magra', '65%'), ('2019-08-30 09:01:08', 6,
'Massa grassa', '35%'),
('2019-09-09 08:54:38', 6, 'Livello idratazione', '87%'), ('2019-09-09
08:54:38', 6, 'Livello deambulazione', 'Medio'), ('2019-09-09 08:54:38', 6,
'Livello vigilanza', 2), ('2019-09-09 08:54:38', 6, 'Risposta oculare',
'Bassa'),
('2019-04-25 15:40:40', 10, 'Livello vigilanza', 4), ('2019-04-25 15:40:40',
10, 'Funzionalità fegato', 'Ok'), ('2019-04-25 15:40:40', 10, 'Funzionalità
cuore', 'Ok'), ('2019-04-25 15:40:40', 10, 'Funzionalità pancreas', 'Ok'),
('2019-09-23 13:14:09', 14, 'Livello deambulazione', 'Medio'), ('2019-09-23
13:14:09', 14, 'Risposta oculare', 'Bassa'), ('2019-09-23 13:14:09', 14,
'Livello idratazione', '60%'),
('2018-11-13 16:10:28', 18, 'Livello vigilanza', 5), ('2018-11-13 16:10:28',
18, 'Livello deambulazione', 'Ottimo'), ('2018-11-13 16:10:28', 18,
'Lucentezza pelo', 'Lucente'), ('2018-11-13 16:10:28', 18, 'Respirazione',
'Corretta'), ('2018-11-13 16:10:28', 18, 'Livello idratazione', '90%'),
('2019-06-01 18:56:01', 22, 'Massa magra', '50%'), ('2019-06-01 18:56:01',
22, 'Massa grassa', '50%'), ('2019-06-01 18:56:01', 22, 'Funzionalità
cuore', 'Ok'), ('2019-06-01 18:56:01', 22, 'Funzionalità pancreas', 'Ok'),
('2016-02-21 16:28:14', 23, 'Massa magra', '40%'), ('2016-02-21 16:28:14',
23, 'Massa grassa', '60%'), ('2016-02-21 16:28:14', 23, 'Spessore zoccolo',
8), ('2016-02-21 16:28:14', 23, 'Livello deambulazione', 'Ottimo');

```

```

INSERT INTO `Terapia` (DataInizio, Durata, Animale) VALUES

```

```
( '2019-09-18', 6, 1),
( '2019-08-30', 10, 6),
( '2019-09-09', 10, 6),
( '2019-09-19', 25, 6),
( '2019-08-29', 25, 14),
( '2018-11-13', 18, 18),
( '2015-12-12', 5, 16),
( '2018-08-15', 10, 18);
```

```
INSERT INTO `Comporta` VALUES
(6, 5),
(7, 6);
```

```
INSERT INTO `Esordio` VALUES
(1, '2019-09-18 16:10:28', 1, 'Gastroenterotossiemia', 5),
(2, '2019-08-30 09:01:08', 6, 'Gastroenterotossiemia', 4),
(3, '2019-09-09 08:54:38', 6, 'Gastroenterotossiemia', 6),
(4, '2019-09-19 10:12:06', 6, 'Gastroenterotossiemia', 9),
(5, '2019-08-29 16:32:21', 14, 'Febbre Q', 4),
(6, '2018-11-13 16:10:28', 18, 'Brucellosi', 7);
```

```
INSERT INTO `Somministra` VALUES
(1, 'Normix', 0),
(2, 'Normix', 0),
(3, 'Normix', 0),
(4, 'Normix', 0),
(5, 'Ciproxin', 0),
(6, 'Cotrimossazolo', 0),
(6, 'Rifadin', 0);
```

```
INSERT INTO `Disturbo` VALUES
('Encefalopatia spongiforme bovina', '2019-07-19 17:08:31', 2, 'Lieve'),
('Encefalopatia spongiforme bovina', '2018-12-28 12:53:58', 3, 'Moderata'),
('Encefalopatia spongiforme bovina', '2018-09-28 16:58:08', 16, 'Lieve');
```

```
INSERT INTO `Lesione` VALUES
('Erosione', 'Fianco dx', '2019-04-25 15:40:40', 10, 'Lieve'),
('Erosione', 'Arto posteriore dx', '2019-04-28 15:27:29', 11, 'Lieve'),
('Escoriazione', 'Mandibola lato sx', '2018-05-22 08:31:33', 8, 'Moderata'),
('Attintura', 'Zoccolo posteriore dx', '2015-07-10 12:34:39', 20, 'Grave');
```

```
INSERT INTO `Cliente` VALUES
('PI5448962K', 'Sibilla', 'Verdiani', 1, '5108136851218544'),
('LI6333860L', 'Sara', 'Arancini', 0, '4148137874316834'),
('TO3317888L', 'Pia', 'Grigini', 1, '5118135662657143'),
('RM7343932I', 'Marco', 'Cianini', 1, '5109133446715376'),
('AE43521395', 'Lorenzo', 'Lino', 0, '5108137864561555'),
('IU47292048', 'Pio', 'Mogani', 0, '5108137754774455'),
('UA23850380', 'Nuccia', 'Ocri', 0, '5108138428576733'),
('EA59029492', 'Dario', 'Platini', 0, '5108134337882164'),
('EU38403840', 'Remigio', 'Magenti', 0, '5108134218376768'),
('OI33948202', 'Laura', 'Ocreni', 0, '5108132585385221'),
('GU38294829', 'Beato', 'Ori', 0, '5108137275658753'),
('PB38434832', 'Anita', 'Pistacchi', 0, '5108134131524828'),
('QA19384920', 'Cupido', 'Porporetta', 0, '5108138433188867'),
('UA39493948', 'Maria', 'Rubini', 0, '5108135686154873'),
('EA49029402', 'Lazzaro', 'Turchesi', 0, '5108132487325673'),
('PL38293832', 'Marisa', 'Argentoni', 0, '5108131423488485'),
('SE48029302', 'Terzo', 'Zafferani', 0, '5108137881161762'),
('ZO29032029', 'Flaminio', 'Bluni', 0, '5108134162518187');
```

```
('LH39202302', 'Giusta', 'Neri', 0, '5108136512734317'),
('DE33492049', 'Zaccaria', 'Aranci', 0, '5108132167516472');
```

```
INSERT INTO `Account` VALUES
('PI5448962K', 'Sibilla', 'J7fCcgo85v', 'PRMSLL55T52C243B', 'Via Saredgna 3,
Lecce (LE)', '3732647603', '2017-09-16', 'Dove sono nata?', 'Castelnuovo
scrivia'),
('RM7343932I', 'Pia', 'UnpjPealCK', 'CSSPIA52C63B932C', 'Via Champigny
48, Rosignano Marittimo (LI)', '3701821476', '2019-08-02', 'Come si chiama
mia mamma?', 'Cinzia'),
('TO3317888L', 'Marco97', 'emipJeQNcV', 'RRTNR97B22A097F', 'Via Nazionale
111, Gioia Tauro (RC)', '3637203258', '2018-03-29', 'Qual e' la mia squadra
di calcio preferita?', 'Inter');
```

```
INSERT INTO `Documento` VALUES
('PI5448962K', 'Patente', '2025-12-12', 'Motorizzazione'),
('RM7343932I', 'Patente', '2022-03-23', 'Motorizzazione'),
('TO3317888L', 'Patente', '2027-02-22', 'Motorizzazione');
```

```
INSERT INTO `Camera` (Numero, Agriturismo, Tipo, Costo) VALUES
(1, 1, 'Semplice', 60), (2, 1, 'Suite', 90), (3, 1, 'Semplice', 30),
(1, 2, 'Semplice', 60), (2, 2, 'Semplice', 60), (3, 2, 'Semplice', 30),
(1, 3, 'Semplice', 60), (2, 3, 'Semplice', 60), (3, 3, 'Suite', 160),
(1, 4, 'Semplice', 30), (2, 4, 'Semplice', 30), (3, 4, 'Semplice', 60),
(1, 5, 'Semplice', 30), (2, 5, 'Semplice', 30), (3, 5, 'Semplice', 60),
(1, 6, 'Suite', 160), (2, 6, 'Semplice', 30), (3, 6, 'Semplice', 60), (4, 6,
'Semplice', 30), (5, 6, 'Semplice', 60);
```

```
INSERT INTO `Capienza` VALUES
('Singolo', 1),
('Matrimoniale', 2);
```

```
INSERT INTO `Letto` (Tipo, Camera, Agriturismo) VALUES
('Matrimoniale', 1, 1),
('Singolo', 2, 1), ('Singolo', 2, 1),
('Singolo', 3, 1),
('Matrimoniale', 1, 2),
('Matrimoniale', 2, 2),
('Singolo', 3, 2),
('Matrimoniale', 1, 3),
('Matrimoniale', 2, 3),
('Matrimoniale', 3, 3), ('Singolo', 3, 3), ('Singolo', 3, 3),
('Singolo', 1, 4),
('Singolo', 2, 4),
('Matrimoniale', 3, 4),
('Singolo', 1, 5),
('Singolo', 2, 5),
('Matrimoniale', 3, 5),
('Matrimoniale', 1, 6), ('Matrimoniale', 1, 6),
('Singolo', 2, 6),
('Matrimoniale', 3, 6),
('Singolo', 4, 6),
('Matrimoniale', 5, 6);
```

```
INSERT INTO `Prenotazione` (DataPartenza, DataArrivo, Cliente) VALUES
('2019-07-06', '2019-07-02', 'PI5448962K'),
('2018-04-10', '2018-03-29', 'LI6333860L'),
('2018-10-15', '2018-10-07', 'TO3317888L'),
('2019-10-01', '2019-09-26', 'LI6333860L'),
('2019-10-02', '2019-09-26', 'PI5448962K'),
```

```

('2019-10-01', '2019-09-26', 'AE43521395'),
('2019-09-30', '2019-09-26', 'IU47292048'),
('2019-10-05', '2019-09-26', 'UA23850380'),
('2019-09-29', '2019-09-26', 'EA59029492'),
('2019-10-02', '2019-09-26', 'EU38403840'),
('2019-10-03', '2019-09-26', 'OI33948202'),
('2019-10-05', '2019-09-26', 'GU38294829'),
('2019-09-30', '2019-09-26', 'PB38434832'),
('2019-10-08', '2019-10-02', 'QA19384920'),
('2019-09-30', '2019-09-26', 'UA39493948'),
('2019-09-30', '2019-09-26', 'EA49029402'),
('2019-10-10', '2019-09-26', 'PL38293832'),
('2019-10-03', '2019-09-26', 'SE48029302'),
('2019-10-04', '2019-09-26', 'ZO29032029'),
('2019-10-04', '2019-09-26', 'LH39202302'),
('2019-09-29', '2019-09-26', 'DE33492049'),
('2019-10-02', '2019-09-24', 'TO3317888L');

```

```
INSERT INTO `Assegnazione` VALUES
```

```

(1, 1, 1), (1, 2, 1),
(2, 1, 3),
(3, 1, 6),
(4, 1, 3),
(5, 1, 4),
(6, 2, 1),
(7, 1, 2),
(8, 1, 5),
(9, 1, 6),
(10, 2, 2),
(11, 2, 3),
(12, 2, 5),
(13, 2, 6),
(14, 2, 1),
(15, 2, 4),
(16, 3, 4),
(17, 3, 3),
(18, 3, 2),
(19, 3, 1),
(20, 3, 5),
(21, 3, 6), (21, 4, 6),
(22, 5, 6);

```

```
INSERT INTO `Pagamento` (Istante, MetodoPagamento, Importo, Prenotazione)
VALUES
```

```

('2018-04-10 11:13:48', 'Carta', 50, 2), ('2018-03-27 00:25:56', 'Carta',
30, 2),
('2018-10-15 08:47:01', 'Contanti', 215, 3),
('2019-07-06 10:00:09', 'Carta', 220, 1),
('2019-09-25 00:12:46', 'Carta', 15, 5),
('2019-09-24 22:42:01', 'Carta', 45, 6),
('2019-08-20 20:21:23', 'Carta', 30, 7),
('2019-08-23 10:34:33', 'Carta', 15, 8),
('2019-09-10 08:14:09', 'Carta', 80, 9),
('2019-07-20 11:19:09', 'Carta', 30, 10),
('2019-08-20 23:12:46', 'Carta', 30, 11),
('2019-09-02 16:32:01', 'Carta', 15, 12),
('2019-09-03 17:33:23', 'Carta', 15, 13),
('2019-08-29 19:45:43', 'Carta', 45, 14),
('2019-09-15 09:17:19', 'Carta', 15, 15),
('2019-09-09 08:01:08', 'Carta', 30, 16),

```

```

('2019-09-06 11:00:46', 'Carta', 80, 17),
('2019-08-31 22:27:21', 'Carta', 15, 18),
('2019-07-30 01:01:07', 'Carta', 15, 19),
('2019-09-19 15:16:30', 'Carta', 30, 20),
('2019-09-18 08:59:59', 'Carta', 45, 21);

```

```

INSERT INTO `Guida` (Nome, Cognome, Agriturismo) VALUES
('Marta', 'Bianchi', 1),
('Ada', 'Neri', 1),
('Luigi', 'Rossi', 2),
('Mario', 'Verdi', 3),
('Viola', 'Bianchi', 3),
('Margherita', 'Celesti', 4),
('Lorella', 'Arancioni', 4),
('Maria', 'Verdolini', 4),
('Giulio', 'Ciano', 5),
('Fabiana', 'Blu', 5),
('Luciana', 'Marrone', 6),
('Fabiola', 'Turchesi', 6),
('Ugo', 'Gialli', 6);

```

```

INSERT INTO `AreaTenuta` (Superficie, Agriturismo) VALUES
(180, 1), (80, 1),
(160, 2),
(230, 3), (80, 3), (110, 3),
(190, 4),
(200, 5),
(120, 6), (150, 6);

```

```

INSERT INTO `Escursione` (OrarioInizio, Costo, Guida) VALUES
('2019-07-05 10:00:00', 25, 1),
('2019-07-04 15:00:00', 30, 1),
('2019-09-20 09:00:00', 19, 1),
('2019-08-03 16:30:00', 30, 3),
('2018-04-05 16:30:00', 20, 4),
('2019-09-27 14:30:00', 22, 6),
('2019-10-29 08:00:00', 29, 9),
('2018-04-05 10:15:00', 30, 11),
('2019-09-26 15:30:00', 15, 12);

```

```

INSERT INTO `Step` VALUES
(1, 1, 20, 1),
(1, 2, 15, 2),
(1, 3, 15, 1), (2, 3, 10, 2),
(1, 4, 15, 3),
(1, 5, 15, 4), (2, 5, 10, 5), (3, 5, 10, 6), (4, 5, 15, 4),
(1, 6, 15, 7),
(1, 7, 20, 8),
(1, 8, 15, 9), (2, 8, 15, 10),
(1, 9, 20, 9);

```

```

INSERT INTO `Prenota` VALUES
(1, 1), (1, 2),
(2, 5),
(22, 9);

```

```

INSERT INTO `ServizioAggiuntivo` VALUES
('Accesso idromassaggio/sauna', 10),
('Accesso spa', 15),
('Accesso palestra', 8),

```



```

('Accesso piscina', 5);

INSERT INTO `Richiede` VALUES
(1, 'Accesso spa', 1),
(3, 'Accesso piscina', 3), (3, 'Accesso palestra', 5),
(22, 'Accesso spa', 2);

INSERT INTO `Condizionatore` (Classe, Locale) VALUES
('B', 1),
('A+', 2),
('A+', 3),
('A', 4),
('A', 5),
('B', 6),
('B', 7),
('A+', 8),
('A', 9),
('B', 10);

INSERT INTO `DispositivoIlluminazione` (GradoProtezione, Lumen, Locale)
VALUES
('IP65', 18000, 1),
('IP65', 20000, 2),
('IP65', 19000, 3),
('IP66', 17000, 3),
('IP65', 18500, 4),
('IP65', 18000, 5),
('IP65', 18000, 6),
('IP65', 20000, 7),
('IP65', 17000, 8),
('IP65', 18000, 9),
('IP65', 20000, 10);

INSERT INTO `Finestra` (Orientamento, Locale) VALUES
('Ovest', 1), ('Est', 1),
('Ovest', 2), ('Est', 2),
('Ovest', 3), ('Est', 3),
('Ovest', 4), ('Est', 4),
('Ovest', 5), ('Est', 5),
('Ovest', 6), ('Est', 6),
('Ovest', 7), ('Est', 7),
('Ovest', 8), ('Est', 8),
('Ovest', 9), ('Est', 9),
('Ovest', 10), ('Est', 10);

INSERT INTO `Foraggio` VALUES
('Meadow hay', 15, 70, 15, 30000, 30, 200, 240, 'Fieno'),
('Meadow pasture', 20, 60, 20, 28000, 25, 250, 210, 'Insilato');

INSERT INTO `Mangiatoia` (Lunghezza, Larghezza, Locale) VALUES
(2, 4, 1), (2, 4, 1),
(3, 2, 2),
(3, 2, 3),
(3, 4, 4),
(3, 2, 5),
(3, 3, 6),
(3, 3, 7),
(3, 2, 8),
(3, 3, 9), (2, 2, 9),
(3, 2, 10);

```

```

INSERT INTO `StatoMangiatoia` VALUES
('2019-09-24 12:12:33', 1, 34, 'Meadow hay'),
('2019-09-24 13:32:09', 2, 39, 'Meadow hay'),
('2019-09-24 18:34:01', 3, 29, 'Meadow hay'),
('2019-09-24 12:02:00', 4, 45, 'Meadow pasture'),
('2019-09-23 18:56:11', 5, 28, 'Meadow hay'),
('2019-09-23 15:05:21', 6, 41, 'Meadow pasture'),
('2019-09-24 16:22:39', 7, 14, 'Meadow pasture'),
('2019-09-23 12:04:16', 8, 40, 'Meadow pasture'),
('2019-09-24 20:51:54', 9, 12, 'Meadow pasture'),
('2019-09-22 14:48:00', 10, 23, 'Meadow hay'),
('2019-09-24 18:49:19', 11, 17, 'Meadow hay'),
('2019-09-24 15:00:52', 12, 2, 'Meadow hay');

```

```

INSERT INTO `Formaggio` VALUES
('Pecorino Romano', 'Toscana', 'Dura', 'Media', 2.6),
('Pecorino Toscano', 'Toscana', 'Dura', 'Media', 3.6),
('Fiore Sardo', 'Sardegna', 'Dura', 'Media', 6.1),
('Crescenza', 'Lombardia', 'Molle', 'Alta', 2.1);

```

```

INSERT INTO `Fase` VALUES
(1, 'Pecorino Romano', 'Al latte crudo o termizzato viene aggiunto lo
scottainnesto proveniente dalla lavorazione precedente.'),
(2, 'Pecorino Romano', 'La coagulazione avviene con caglio di agnello
proveniente dal territorio di produzione.'),
(3, 'Pecorino Romano', 'La cagliata viene tagliata alle dimensioni di un
chicco di riso e successivamente semicotta o cotta.'),
(4, 'Pecorino Romano', 'Dopo l'estrazione e la pressatura, si salano le
forme, a secco o in salamoia.'),
(5, 'Pecorino Romano', 'Al termine il formaggio viene posizionato nelle
cantine per il processo di stagionatura.'),
(1, 'Pecorino Toscano', 'Il latte crudo o pastorizzato viene inoculato con
fermenti lattici naturali o selezionati.'),
(2, 'Pecorino Toscano', 'La coagulazione avviene con caglio di vitello.'),
(3, 'Pecorino Toscano', 'In seguito, la massa viene semicotta, per poi
essere estratta, posta in fuscelle e lasciata in stufatura, previa
pressatura manuale.'),
(4, 'Pecorino Toscano', 'Al termine il formaggio viene posizionato nelle
cantine per il processo di stagionatura.'),
(1, 'Fiore Sardo', 'Il latte crudo viene riscaldato in caldaie di rame.'),
(2, 'Fiore Sardo', 'La coagulazione avviene con caglio di vitello.'),
(3, 'Fiore Sardo', 'La cagliata viene rotta con la chiova alle dimensioni di
un chicco di miglio e lasciata riposare.'),
(4, 'Fiore Sardo', 'Dopo l'estrazione, trova posto in formelle tronco-
coniche, dove le forme subiscono alcuni rivoltamenti che agevolano lo
spurgo.'),
(5, 'Fiore Sardo', 'Al termine il formaggio viene posizionato nelle cantine
per il processo di stagionatura.'),
(1, 'Crescenza', 'Il latte pastorizzato viene inoculato con lattoinnesto e
addizionato con caglio di vitello liquido.'),
(2, 'Crescenza', 'La cagliata viene rotta in due fasi e poi estratta e
inserita in stampi, dove subisce una stufatura. La salatura è in
salamoia.'),
(3, 'Crescenza', 'Al termine il formaggio viene collocato in celle
frigorifere.');
```

```

INSERT INTO `ParametroProcesso` VALUES
('Durata', 1, 'Pecorino Romano', 10),
('Durata', 2, 'Pecorino Romano', 80),

```

```

('Durata', 3, 'Pecorino Romano', 120),
('Temperatura di cottura', 3, 'Pecorino Romano', 45),
('Durata', 4, 'Pecorino Romano', 60),
('Tempi stagionatura', 5, 'Pecorino Romano', 150),
('Durata', 1, 'Pecorino Toscano', 20),
('Durata', 2, 'Pecorino Toscano', 60),
('Durata', 3, 'Pecorino Toscano', 180),
('Tempi stagionatura', 4, 'Pecorino Toscano', 180),
('Durata', 1, 'Fiore Sardo', 40),
('Temperatura del latte', 1, 'Fiore Sardo', 35),
('Durata', 2, 'Fiore Sardo', 60),
('Durata', 3, 'Fiore Sardo', 180),
('Tempi di riposo', 3, 'Fiore Sardo', 5),
('Durata', 4, 'Fiore Sardo', 120),
('Tempi stagionatura', 5, 'Fiore Sardo', 110),
('Durata', 1, 'Crescenza', 70),
('Temperatura caglio', 1, 'Crescenza', 36),
('Durata', 2, 'Crescenza', 120),
('Intervallo tra rottura', 2, 'Crescenza', 30);

INSERT INTO `Mungitrice` (Latitudine, Longitudine) VALUES
(43.58751, 11.31732),
(41.72115, 12.67683),
(45.07250, 7.52732),
(43.26975, 10.59462),
(40.38023, 15.54137),
(43.98369, 12.42408);

INSERT INTO `Silos` (LivelloRiempimento, CapacitaEttolitri, Agriturismo)
VALUES
(0, 1500, 1),
(0, 1200, 2),
(0, 1300, 3),
(0, 1250, 4),
(0, 800, 5),
(0, 800, 5),
(0, 1500, 6);

INSERT INTO `Latte` (Volume, Mungitrice, Animale, Silos) VALUES
(6, 1, 2, 1), (8, 1, 2, 1),
(7, 1, 3, 1), (9, 1, 3, 1),
(8, 2, 5, 2),
(5, 6, 10, 7),
(8, 3, 12, 3),
(7, 5, 14, 5), (8, 5, 14, 6), (5, 5, 14, 6), (10, 5, 14, 5),
(8, 3, 16, 3), (6, 3, 16, 3),
(10, 6, 18, 7), (6, 6, 18, 7), (12, 6, 18, 7),
(9, 1, 20, 1),
(7, 1, 21, 1),
(6, 3, 23, 3);

INSERT INTO `Caratteristica` VALUES
('Carboidrati', 1, 47.8), ('Proteine', 1, 31.5), ('Grassi', 1, 32.7),
('Colesterolo', 1, 0.05), ('Sodio', 1, 0.44), ('Potassio', 1, 1.5),
('Carboidrati', 2, 46.7), ('Proteine', 2, 33), ('Grassi', 2, 31.7),
('Colesterolo', 2, 0.07), ('Sodio', 2, 0.43), ('Potassio', 2, 1.4),
('Carboidrati', 3, 47), ('Proteine', 3, 31.2), ('Grassi', 3, 32.6),
('Colesterolo', 3, 0.08), ('Sodio', 3, 0.42), ('Potassio', 3, 1.3),
('Carboidrati', 4, 47.2), ('Proteine', 4, 32.7), ('Grassi', 4, 31.5),
('Colesterolo', 4, 0.09), ('Sodio', 4, 0.41), ('Potassio', 4, 1.2),

```

```

('Carboidrati', 5, 45.1), ('Proteine', 5, 29.1), ('Grassi', 5, 32.4),
('Colesterolo', 5, 0.10), ('Sodio', 5, 0.4), ('Potassio', 5, 1.1),
('Carboidrati', 6, 44.1), ('Proteine', 6, 31.6), ('Grassi', 6, 30.3),
('Colesterolo', 6, 0.06), ('Sodio', 6, 0.45), ('Potassio', 6, 1.6),
('Carboidrati', 7, 49), ('Proteine', 7, 30.1), ('Grassi', 7, 30.2),
('Colesterolo', 7, 0.05), ('Sodio', 7, 0.46), ('Potassio', 7, 1.7),
('Carboidrati', 8, 47.9), ('Proteine', 8, 31.9), ('Grassi', 8, 32.1),
('Colesterolo', 8, 0.04), ('Sodio', 8, 0.47), ('Potassio', 8, 1.8),
('Carboidrati', 9, 43.1), ('Proteine', 9, 29), ('Grassi', 9, 30),
('Colesterolo', 9, 0.03), ('Sodio', 9, 0.48), ('Potassio', 9, 1.5),
('Carboidrati', 10, 45.6), ('Proteine', 10, 28.9), ('Grassi', 10, 30.9),
('Colesterolo', 10, 0.02), ('Sodio', 10, 0.49), ('Potassio', 10, 1.9),
('Carboidrati', 11, 43.7), ('Proteine', 11, 32.2), ('Grassi', 11, 29.8),
('Colesterolo', 11, 0.03), ('Sodio', 11, 0.43), ('Potassio', 11, 1.3),
('Carboidrati', 12, 49.9), ('Proteine', 12, 32.2), ('Grassi', 12, 31.7),
('Colesterolo', 12, 0.04), ('Sodio', 12, 0.44), ('Potassio', 12, 1.4),
('Carboidrati', 13, 42.3), ('Proteine', 13, 30.4), ('Grassi', 13, 32.6),
('Colesterolo', 13, 0.05), ('Sodio', 13, 0.45), ('Potassio', 13, 1.6),
('Carboidrati', 14, 47.4), ('Proteine', 14, 28.9), ('Grassi', 14, 29.5),
('Colesterolo', 14, 0.07), ('Sodio', 14, 0.48), ('Potassio', 14, 1.7),
('Carboidrati', 15, 46.8), ('Proteine', 15, 29.2), ('Grassi', 15, 32.4),
('Colesterolo', 15, 0.06), ('Sodio', 15, 0.49), ('Potassio', 15, 1.8),
('Carboidrati', 16, 45.5), ('Proteine', 16, 32), ('Grassi', 16, 30.3),
('Colesterolo', 16, 0.08), ('Sodio', 16, 0.44), ('Potassio', 16, 1.2),
('Carboidrati', 17, 46.6), ('Proteine', 17, 32.8), ('Grassi', 17, 30.2),
('Colesterolo', 17, 0.09), ('Sodio', 17, 0.46), ('Potassio', 17, 1.4),
('Carboidrati', 18, 49.2), ('Proteine', 18, 31.7), ('Grassi', 18, 31.1),
('Colesterolo', 18, 0.03), ('Sodio', 18, 0.47), ('Potassio', 18, 1.6),
('Carboidrati', 19, 43.3), ('Proteine', 19, 30.3), ('Grassi', 19, 32),
('Colesterolo', 19, 0.04), ('Sodio', 19, 0.42), ('Potassio', 19, 1.9);

```

```

INSERT INTO `Cantina` (ClasseEnergetica, Agriturismo) VALUES

```

```

('A2', 1),
('A2', 2),
('A1', 3),
('A1', 4),
('A1', 5),
('A3', 6);

```

```

INSERT INTO `StatoCantina` VALUES

```

```

('2019-09-24 16:22:39', 1, 'Forte', 10, 90),
('2019-09-23 12:04:16', 2, 'Leggera', 11, 90),
('2019-09-24 20:51:54', 3, 'Spenta', 12, 86),
('2019-09-22 14:48:00', 4, 'Forte', 11, 80),
('2019-09-24 18:49:19', 5, 'Leggera', 10, 88),
('2019-09-24 15:00:52', 6, 'Forte', 13, 80);

```

```

INSERT INTO `Magazzino` (Superficie, ClasseEnergetica, Agriturismo) VALUES

```

```

(100, 'A1', 1), (60, 'A1', 1),
(90, 'A1', 2),
(120, 'A2', 3),
(110, 'A1', 4),
(90, 'A1', 5),
(130, 'A3', 6), (80, 'A1', 6);

```

```

INSERT INTO `Scaffalatura` (Marca, Portata, CapacitaFormaggi, Magazzino,
Cantina) VALUES

```

```

('EMAF', 3000, 50, 1, NULL),
('EMAF', 2800, 40, 2, NULL),
('EMAF', 3200, 55, 3, NULL),

```

```

('EMAF', 2750, 40, 4, NULL),
('EMAF', 2950, 50, 5, NULL),
('EMAF', 3000, 50, 6, NULL),
('EMAF', 3100, 60, 7, NULL),
('EMAF', 2900, 45, 8, NULL),
('Brescancin', 2750, 40, NULL, 1),
('Brescancin', 3000, 50, NULL, 2),
('Brescancin', 2900, 55, NULL, 3),
('Arioli Afro', 2950, 50, NULL, 4),
('Arioli Afro', 3100, 50, NULL, 5),
('Arioli Afro', 2850, 60, NULL, 6);

```

```

INSERT INTO `LaboratorioProduzione` (Superficie, NumeroVasche, Agriturismo)
VALUES
(120, 10, 1),
(100, 8, 2),
(90, 8, 3),
(115, 11, 4),
(130, 9, 5),
(120, 9, 6);

```

```

INSERT INTO `Ordine` (Istante, Stato, Cliente) VALUES
('2019-09-20 11:50:07', 'Evaso', 'PI5448962K'),
('2019-09-20 23:21:59', 'Evaso', 'TO3317888L'),
('2019-09-19 18:30:21', 'Spedito', 'RM7343932I'),
('2019-09-25 07:45:43', 'Pendente', 'TO3317888L');

```

```

INSERT INTO `Recensione` (Gradimento, Gusto, Qualita, Conservazione,
CodiceOrdine) VALUES
(4, 5, 4, 4, 1),
(3, 5, 5, 3, 2),
(3, 4, 3, 3, 3);

```

```

INSERT INTO `Attesa` VALUES
(4, 'Pecorino Romano', 2);

```

```

INSERT INTO `CentroSmistamento` (NumeroDipendenti, Latitudine, Longitudine)
VALUES
(21, 43.7874, 11.2499),
(27, 41.9109, 12.4818),
(30, 45.0781, 7.6761),
(32, 43.5436, 10.317),
(28, 40.6642, 14.8046),
(25, 44.0693, 12.5566);

```

```

INSERT INTO `Spedizione` (Stato, DataConsegnaPrevista,
DataConsegnaEffettiva, CodiceOrdine) VALUES
('Consegnata', '2019-09-20', '2019-09-20', 1),
('Consegnata', '2019-09-22', '2019-09-23', 2),
('In consegna', '2019-09-25', '2019-09-30', 3);

```

```

INSERT INTO `Percorso` VALUES
(1, 1), (2, 1), (4, 1),
(2, 2), (4, 2),
(1, 3), (2, 3), (3, 3), (5, 3), (6, 3);

```

```

INSERT INTO `Lotto` (`Data`, Scadenza, DurataProduzione, Silos, Laboratorio)
VALUES
('2019-04-18', '2019-05-18', 260, 1, 1),
('2018-04-17', '2019-05-17', 285, 1, 1),

```

```

('2019-03-30', '2019-10-30', 275, 2, 2),
('2019-03-21', '2019-04-21', 315, 2, 2),
('2019-09-22', '2019-10-06', 205, 3, 3),
('2019-09-25', '2019-10-10', 190, 3, 3),
('2019-08-15', '2020-01-01', 430, 4, 4),
('2019-08-29', '2020-01-13', 395, 5, 5),
('2019-07-09', '2019-11-24', 430, 6, 5),
('2019-09-22', '2020-02-10', 350, 7, 6),
('2019-04-29', '2020-10-26', 280, 7, 6);

```

```

INSERT INTO `ProdottoCaseario` (Peso, Lotto, Formaggio, Scaffalatura,
Ordine) VALUES
(0.352, 1, 'Pecorino Romano', 1, 1), (1.03, 2, 'Pecorino Romano', 2, 2),
(0.52, 3, 'Pecorino Toscano', 10, NULL), (0.54, 4, 'Pecorino Toscano', 3,
3),
(1.3, 5, 'Crescenza', 4, NULL), (1.09, 6, 'Crescenza', 4, NULL),
(1, 7, 'Fiore Sardo', 12, NULL), (0.131, 8, 'Fiore Sardo', 13, NULL), (0.5,
9, 'Fiore Sardo', 13, NULL), (0.87, 10, 'Fiore Sardo', 14, NULL),
(0.3, 11, 'Pecorino Romano', 14, NULL), (0.36, 11, 'Pecorino Romano', 14,
NULL), (1.348, 11, 'Pecorino Romano', 14, NULL);

```

```

INSERT INTO `SchedaValutazione` VALUES
(1, 'Nauseante', 'Normale', 1, 4);

```

```

INSERT INTO `ParametroEffettivo` VALUES
('Durata', 1, 1, 15), ('Durata', 2, 1, 85), ('Durata', 3, 1, 120),
('Temperatura di cottura', 3, 1, 46), ('Durata', 4, 1, 60),
('Durata', 1, 2, 10), ('Durata', 2, 2, 80), ('Durata', 3, 2, 130),
('Temperatura di cottura', 3, 2, 44), ('Durata', 4, 2, 65),
('Durata', 1, 3, 25), ('Durata', 2, 3, 60), ('Durata', 3, 3, 190),
('Durata', 1, 4, 35), ('Durata', 2, 4, 80), ('Durata', 3, 4, 200),
('Durata', 1, 5, 75), ('Temperatura caglio', 1, 5, 36), ('Durata', 2, 5,
130), ('Intervallo tra rottura', 2, 5, 35),
('Durata', 1, 6, 70), ('Temperatura caglio', 1, 6, 37), ('Durata', 2, 6,
120), ('Intervallo tra rottura', 2, 6, 30),
('Durata', 1, 7, 40), ('Temperatura del latte', 1, 7, 38), ('Durata', 2, 7,
80), ('Durata', 3, 7, 180), ('Tempi di riposo', 3, 7, 10), ('Durata', 4, 7,
130),
('Durata', 1, 8, 45), ('Temperatura del latte', 1, 8, 34), ('Durata', 2, 8,
60), ('Durata', 3, 8, 160), ('Tempi di riposo', 3, 8, 5), ('Durata', 4, 8,
130),
('Durata', 1, 9, 40), ('Temperatura del latte', 1, 9, 35), ('Durata', 2, 9,
60), ('Durata', 3, 9, 190), ('Tempi di riposo', 3, 9, 5), ('Durata', 4, 9,
140),
('Durata', 1, 10, 30), ('Temperatura del latte', 1, 10, 35), ('Durata', 2,
10, 50), ('Durata', 3, 10, 160), ('Tempi di riposo', 3, 10, 5), ('Durata',
4, 10, 110),
('Durata', 1, 11, 10), ('Durata', 2, 11, 80), ('Durata', 3, 11, 130),
('Temperatura di cottura', 3, 11, 45), ('Durata', 4, 11, 60);

```

```

INSERT INTO `Operaio` (Stipendio, Nome, Cognome) VALUES
(1300, 'Giuseppe', 'Marretti'),
(1350, 'Marta', 'Nera'),
(1200, 'Chiara', 'Rossone'),
(1200, 'Vera', 'Biancani'),
(1100, 'Gabriele', 'Arancina'),
(1000, 'Alessandro', 'Gialletti'),
(900, 'Lucia', 'Vaniglia'),
(1500, 'Monica', 'Ambra'),
(1300, 'Elena', 'Amaranti'),

```

```
(1200, 'Lara', 'Cobalti'),
(1200, 'Lorenzo', 'Indaco'),
(1200, 'Franco', 'Lilla');
```

```
INSERT INTO `Lavora` VALUES
```

```
(1, 1),
(2, 1), (2, 2),
(3, 1), (3, 2),
(4, 1), (4, 2),
(5, 3), (5, 4),
(6, 3), (6, 4),
(7, 5),
(8, 6),
(9, 7),
(10, 8), (10, 9),
(11, 8), (11, 9),
(12, 10),
(12, 11);
```

```
-- -----
--                                     --
--                               Analytics --
--                                     --
-- -----
```

```
CREATE TABLE IF NOT EXISTS MV_ATTIVITA (
    Animale INT UNSIGNED NOT NULL,
    NumeroSpostamenti INT UNSIGNED NOT NULL,
    ScostamentoPerc FLOAT UNSIGNED NOT NULL,
    TipoScostamento VARCHAR(30) NOT NULL,
    PRIMARY KEY(Animale)
);
```

```
INSERT INTO MV_ATTIVITA
```

```
-- Ottengo lo scostamento % tra gli spostamenti di ogni animale e la
media
```

```
-- degli spostamenti degli animali dello stesso agriturismo
```

```
SELECT D2.Animale,
       D2.Spostamenti,
       ROUND(ABS(D2.SpostamentiMedi - D2.Spostamenti) * 100 /
D2.SpostamentiMedi, 2), -- Scostamento
       IF (D2.SpostamentiMedi - D2.Spostamenti < 0, 'Spostamenti
sopra la media', 'Spostamenti sotto la media') -- TipoScostamento
FROM (
```

```
-- Numero medio di volte che gli animali di uno stesso
agriturismo si spostano durante il pascolo
```

```
SELECT D.Animale,
       D.Spostamenti,
       AVG(D.Spostamenti) OVER(PARTITION BY
S.Agriturismo) AS SpostamentiMedi
FROM (
```

```
-- Conto il numero di volte che un animale si
sposta
```

```
SELECT Animale, COUNT(DISTINCT latitudine,
longitudine) - 1 AS Spostamenti
```

```
FROM Posizione
WHERE Istante > CURRENT_DATE - INTERVAL 7 DAY --
Posizioni di "questa settimana"
```

```
GROUP BY Animale
```

```
) AS D
```

```
INNER JOIN
```

```

        Animale A ON D.Animale = A.Codice
        INNER JOIN
        Locale L ON L.Codice = A.Locale
        INNER JOIN
        Stalla S ON S.Codice = L.Stalla
    ) AS D2;

-- Deferred refresh
DELIMITER $$
DROP EVENT IF EXISTS Refresh_MV_Attivita $$
CREATE EVENT Refresh_MV_Attivita ON SCHEDULE EVERY 1 DAY
STARTS '2019-01-01 02:00:00'
DO
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        SELECT 'Si è verificato un errore, materialized view non
aggiornata.';
    END;

    TRUNCATE TABLE MV_ATTIVITA;

    INSERT INTO MV_ATTIVITA
    -- Ottengo lo scostamento % tra gli spostamenti di ogni animale e la
media
    -- degli spostamenti degli animali dello stesso agriturismo
    SELECT D2.Animale,
           D2.Spostamenti,
           ROUND(ABS(D2.SpostamentiMedi - D2.Spostamenti) * 100 /
D2.SpostamentiMedi, 2), -- Scostamento
           IF (D2.SpostamentiMedi - D2.Spostamenti < 0, 'Spostamenti
sopra la media', 'Spostamenti sotto la media') -- TipoScostamento
    FROM (
        -- Numero medio di volte che gli animali di uno stesso
agriturismo si spostano durante il pascolo
        SELECT D.Animale,
               D.Spostamenti,
               AVG(D.Spostamenti) OVER(PARTITION BY
S.Agriturismo) AS SpostamentiMedi
        FROM (
            -- Conto il numero di volte che un animale si
sposta
            SELECT Animale, COUNT(DISTINCT latitudine,
longitudine) - 1 AS Spostamenti
            FROM Posizione
            WHERE Istante > CURRENT_DATE - INTERVAL 7 DAY --
Posizioni di "questa settimana"
            GROUP BY Animale
        ) AS D
        INNER JOIN
        Animale A ON D.Animale = A.Codice
        INNER JOIN
        Locale L ON L.Codice = A.Locale
        INNER JOIN
        Stalla S ON S.Codice = L.Stalla
    ) AS D2;
END $$
DELIMITER ;

```



```

CREATE TABLE IF NOT EXISTS MV_CORTEGGIAMENTO (
    IstanteRilevazione1 TIMESTAMP NOT NULL, -- Istante di rilevazione
della posizione
    Animale1 INT UNSIGNED NOT NULL,
    Sesso1 CHAR(1) NOT NULL,
    IstanteRilevazione2 TIMESTAMP NOT NULL, -- Istante di rilevazione della
posizione
    Animale2 INT UNSIGNED NOT NULL,
    Sesso2 CHAR(1) NOT NULL,
    Distanza FLOAT UNSIGNED NOT NULL, -- Distanza fra i due animali
    PRIMARY KEY (IstanteRilevazione1, Animale1, IstanteRilevazione2,
Animale2)
);

INSERT INTO MV_CORTEGGIAMENTO
    -- Ottengo la distanza tra tutte le posizioni dell'animale che sto
considerando
    -- e tutte le posizioni di tutti gli altri animali
    SELECT P1.Istante, P1.Animale, A1.Sesso, P2.Istante, P2.Animale,
A2.Sesso,
        ROUND(
            111111 -- Numero di metri per grado di
latitudine
            * DEGREES(ACOS(LEAST(1.0,
COS(RADIANS(P1.Latitudine))
            * COS(RADIANS(P2.Latitudine))
            * COS(RADIANS(P1.Longitudine - P2.Longitudine))
            + SIN(RADIANS(P1.Latitudine))
            * SIN(RADIANS(P2.Latitudine))))))
        ) AS Distanza
    FROM Posizione P1
        INNER JOIN
            Animale A1 ON A1.Codice = P1.Animale
        INNER JOIN
            Posizione P2 ON (
                P1.Animale <> P2.Animale
                AND P1.Animale < P2.Animale -- Evita i
"duplicati" del self join
                -- Le rilevazioni devono essere avvenute con uno
scarto massimo di 30 secondi,
                -- altrimenti chiunque potrebbe essere stato
vicino a chiunque nell'arco di giorni e giorni
                -- In altre parole: se calcolo la distanza tra
due posizioni senza considerare il tempo
                -- potrei trovarmi un animale che è stato nella
posizione X il giorno Y e un altro animale
                -- che è stato sempre nel posto X in un giorno
diverso, sembrano che siano stati vicini anche se,
                -- ovviamente, non è stato così.
                AND ABS(TIMESTAMPDIFF(SECOND, P1.Istante,
P2.Istante)) <= 30
            )
        INNER JOIN
            Animale A2 ON A2.Codice = P2.Animale
    WHERE A2.Locale = A1.Locale -- Mi interessa la distanza solo fra
animali di uno stesso locale
        AND A1.Sesso <> A2.Sesso -- Elimina gli animali dello
stesso sesso

```

```

        AND DATE(P1.Istante) > CURRENT_DATE - INTERVAL 7 DAY; --
Posizioni di "questa settimana"

-- Deferred refresh
DELIMITER $$
DROP EVENT IF EXISTS Refresh_MV_Corteggiamento $$
CREATE EVENT Refresh_MV_Corteggiamento
ON SCHEDULE EVERY 1 DAY
STARTS '2019-01-01 02:30:00'
DO
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        SELECT 'Si è verificato un errore, materialized view non
aggiornata.';
    END;

    TRUNCATE TABLE MV_CORTEGGIAMENTO;

    INSERT INTO MV_CORTEGGIAMENTO
    -- Ottengo la distanza tra tutte le posizioni dell'animale che sto
considerando
    -- e tutte le posizioni di tutti gli altri animali
    SELECT P1.Istante, P1.Animale, A1.Sesso, P2.Istante, P2.Animale,
A2.Sesso,
        ROUND(
            111111 -- Numero di metri per grado di
latitudine
            * DEGREES(ACOS(LEAST(1.0,
COS(RADIANS(P1.Latitudine))
            * COS(RADIANS(P2.Latitudine))
            * COS(RADIANS(P1.Longitudine - P2.Longitudine))
            + SIN(RADIANS(P1.Latitudine))
            * SIN(RADIANS(P2.Latitudine)))))
        ) AS Distanza
    FROM Posizione P1
    INNER JOIN
        Animale A1 ON A1.Codice = P1.Animale
    INNER JOIN
        Posizione P2 ON (
            P1.Animale <> P2.Animale
            AND P1.Animale < P2.Animale -- Evita i
"duplicati" del self join
            -- Le rilevazioni devono essere avvenute con uno
scarto massimo di 30 secondi,
            -- altrimenti chiunque potrebbe essere stato
vicino a chiunque nell'arco di giorni e giorni
            -- In altre parole: se calcolo la distanza tra
due posizioni senza considerare il tempo
            -- potrei trovarmi un animale che è stato nella
posizione X il giorno Y e un altro animale
            -- che è stato sempre nel posto X in un giorno
diverso, sembrano che siano stati vicini anche se,
            -- ovviamente, non è stato così.
            AND ABS(TIMESTAMPDIFF(SECOND, P1.Istante,
P2.Istante)) <= 30
        )
    INNER JOIN
        Animale A2 ON A2.Codice = P2.Animale

```

```

WHERE A2.Locale = A1.Locale -- Mi interessa la distanza solo fra
animali di uno stesso locale
      AND A1.Sesso <> A2.Sesso -- Elimina gli animali dello
stesso sesso
      AND DATE(P1.Istante) > CURRENT_DATE - INTERVAL 7 DAY; --
Posizioni di "questa settimana"
END $$
DELIMITER ;

CREATE TABLE IF NOT EXISTS MV_ISOLAMENTO (
  Animale INT UNSIGNED NOT NULL,
  Tipo VARCHAR(10) NOT NULL,
  GiorniIsolamento TINYINT UNSIGNED NOT NULL,
  PRIMARY KEY (Animale)
);

INSERT INTO MV_ISOLAMENTO
  WITH distanze AS (
    -- Per ogni animale rilevo la distanza giornaliera media dagli
    altri animali
    SELECT DATE(P1.Istante) AS `Data`,
           P1.Animale,
           A1.Tipo,
           AVG (
             111111 -- Numero di metri per grado di
             latitudine
             * DEGREES(ACOS(LEAST(1.0,
             COS(RADIANS(P1.Latitudine))
             * COS(RADIANS(P2.Latitudine))
             * COS(RADIANS(P1.Longitudine -
             P2.Longitudine))
             + SIN(RADIANS(P1.Latitudine))
             * SIN(RADIANS(P2.Latitudine))))))
           ) AS DistanzaMedia
    FROM Posizione P1
      INNER JOIN
      Animale A1 ON A1.Codice = P1.Animale
      INNER JOIN
      Posizione P2 ON (
        P1.Animale <> P2.Animale
        -- Le rilevazioni devono essere
        avvenute con uno scarto massimo di 15 secondi,
        -- altrimenti la rilevazione
        dell'animale più vicino potrebbe essere errata
        AND ABS(TIMESTAMPDIFF(SECOND,
        P1.Istante, P2.Istante)) <= 15
      )
      INNER JOIN
      Animale A2 ON A2.Codice = P2.Animale
    WHERE A2.Locale = A1.Locale -- Mi interessa la distanza solo fra
    animali di uno stesso locale
      AND DATE(P1.Istante) > CURRENT_DATE - INTERVAL 7 DAY -
- Posizioni di "questa settimana"
    GROUP BY DATE(P1.Istante), P1.Animale
  )
  -- Conto il numero di giorni in cui ogni animale è stato isolato
  SELECT Animale, Tipo, COUNT(*) -- GiorniIsolamento
  FROM distanze

```

```

WHERE DistanzaMedia >= 6 -- Oltre i 6 metri viene considerato isolato
GROUP BY Animale;

-- Deferred refresh
DELIMITER $$
DROP EVENT IF EXISTS Refresh_MV_Corteggiamento $$
CREATE EVENT Refresh_MV_Corteggiamento
ON SCHEDULE EVERY 1 DAY
STARTS '2019-01-01 03:00:00'
DO
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        SELECT 'Si è verificato un errore, materialized view non
aggiornata.';
    END;

    TRUNCATE TABLE MV_ISOLAMENTO;

    INSERT INTO MV_ISOLAMENTO
    WITH distanze AS (
        -- Per ogni animale rilevo la distanza giornaliera media dagli
altri animali
        SELECT DATE(P1.Istante) AS `Data`,
            P1.Animale,
            A1.Tipo,
            AVG (
latitudine
                                111111 -- Numero di metri per grado di
                                * DEGREES(ACOS(LEAST(1.0,
COS(RADIANS(P1.Latitudine))
                                * COS(RADIANS(P2.Latitudine))
                                * COS(RADIANS(P1.Longitudine -
P2.Longitudine))
                                + SIN(RADIANS(P1.Latitudine))
                                * SIN(RADIANS(P2.Latitudine))))
                                ) AS DistanzaMedia
        FROM Posizione P1
            INNER JOIN
                Animale A1 ON A1.Codice = P1.Animale
            INNER JOIN
                Posizione P2 ON (
                                P1.Animale <> P2.Animale
                                -- Le rilevazioni devono essere
avvenute con uno scarto massimo di 15 secondi,
                                -- altrimenti la rilevazione
dell'animale più vicino potrebbe essere errata
                                AND ABS(TIMESTAMPDIFF(SECOND,
P1.Istante, P2.Istante)) <= 15
                                )
            INNER JOIN
                Animale A2 ON A2.Codice = P2.Animale
        WHERE A2.Locale = A1.Locale -- Mi interessa la distanza solo fra
animali di uno stesso locale
            AND DATE(P1.Istante) > CURRENT_DATE - INTERVAL 7 DAY -
- Posizioni di "questa settimana"
        GROUP BY DATE(P1.Istante), P1.Animale
    )
    -- Conto il numero di giorni in cui ogni animale è stato isolato

```

```
SELECT Animale, Tipo, COUNT(*) -- GiorniIsolamento
FROM distanze
WHERE DistanzaMedia >= 6 -- Oltre i 6 metri viene considerato isolato
GROUP BY Animale;
END $$
DELIMITER ;
```