



ANNO ACCADEMICO 2018 – 2019

Farmhouse 4.0



**LAURETTA DARIO
PISANA MARTINA**





INDICE

1. Visione d'insieme e analisi del database	3
2. Glossario.....	4
2.1 Area Allevamento	4
2.2 Area Healthcare	6
2.3 Area Produzione	7
2.4 Area Soggiorno	8
2.5 Area Store	9
3. Dizionario dei dati	10
3.1 Area Allevamento	10
3.2 Area Healthcare	13
3.3 Area Produzione	15
3.4 Area Soggiorno	17
3.5 Area Store	18
4. Attributi Derivati	19
5. Generalizzazioni	22
6. Attributi Composti.....	24
7. Traduzione delle generalizzazioni ed eliminazione attributi composti / multivalore	26
8. Ristrutturazione del diagramma E/R.....	30
8.1 Tavola dei Volumi	30
8.1.1 Tavola Volumi Area Allevamento	30
8.1.2 Tavola Volumi Area Healthcare	32
8.1.3 Tavola Volumi Area Produzione	33
8.1.4 Tavola Volumi Area Soggiorno	34
8.1.5 Tavola Volumi Area Store	35
8.2 Tavola delle Operazioni	37
8.2.1 Tavola Operazione1	38
8.2.2 Tavola Operazione2	40
8.2.3 Tavola Operazione3	41
8.2.4 Tavola Operazione4	42
8.2.5 Tavola Operazione5	44
8.2.6 Tavola Operazione6	46
8.2.7 Tavola Operazione7	47
8.2.8 Tavola Operazione8	49
9. Traduzione modello logico	51

9.1 Area Allevamento	51
9.2 Area Healthcare	54
9.3 Area Produzione	56
9.4 Area Soggiorno	58
9.5 Area Store	59
10. Dipendenze Funzionali.....	60
11. Trigger gestione attributi derivati	73
12. Business Rule	80
13. Analytics	106
13.1 Comportamento degli animali	106
13.2 Tracciabilità di filiera	109
13.3 Analisi vendite di un prodotto caseario	112

1. Visione d'insieme e analisi del database

L'obiettivo del progetto è la creazione di una base di dati che permetta la memorizzazione e il management dei dati del sistema informativo dell'impresa "Farmhouse 4.0". Essa, si occupa di gestire una rete di agriturismi di avanzata tecnologia, basati sui nuovi principi dell'industria intelligente (Industry 4.0), che svolgeranno un ruolo importante, ovvero quello di aiutare a migliorare il processo decisionale e la produttività. Grazie alla scoperta di queste nuove tecnologie, il rapporto tra essere umano e macchina è cambiato radicalmente, basta pensare all'interazione che vi è tra i robot, che non solo facilitano il lavoro svolto dall'uomo, ma lo rendono anche più sicuro. Uno degli scopi principali di questa grande rete di agriturismi è cercare di ridurre i costi per la manutenzione e gli sprechi, e di garantire il benessere di tutti gli animali



Per agevolarci nella progettazione e nella realizzazione dell'elaborato, si è deciso di suddividere il progetto in 6 aree tematiche, mantenendo la divisione presentata nella documentazione originale:

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

2. Glossario

In questo paragrafo, vengono illustrati i vari glossari, uno per ogni area. Lo scopo del glossario è quello di rendere più semplice la comprensione dei termini usati.

2.1 Area Allevamento

L'area allevamento contiene tutte le informazioni riguardo la gestione dei vari allevamenti di FarmHouse 4.0.

Termine	Descrizione	Sinonimi	Collegamenti
Animale	Animale che risiede nelle stalle. Può essere un bovino, un caprino o un ovino.		Fornitore, locale, riproduzione, Gestazione, Esame, Terapia, Patologia, Complicanza, Gravidanza, Visita, Posizione (GPS), Orario di rientro, Scheda Medica, Indicatori, Mungitura
Fornitore	Il fornitore vende gli animali.	Venditore, negoziante, commerciante	Animale
Stalla	La stalla racchiude più locali.		Locale
Locale	Luogo in cui risiedono gli animali della stessa specie. Ogni locale ha un proprio allestimento.	Ambiente, vano, stanza	Stalla, animale, allestimento, livello sporcizia, parametri locale, Attività di Pascolo, Mungitura
Allestimento	Gli allestimenti dipendono dalla specie e prevedono un numero e una tipologia di mangiatoie, abbeveratoi, ecc...	Organizzazione, disposizione, preparazione, sistemazione	Locale, mangiatoia, abbeveratoio
Mangiatoia	È dotata di un sensore che registra la quantità e il tipo di foraggio contenuto.	Greppia	Allestimento, foraggio

Abbeveratoio	È dotato di un sensore che registra la quantità di acqua contenuta.		Allestimento
Foraggio	Tipo di pasto di una determinata specie. Per ogni foraggio sono specificate le diverse componenti.		Mangiatoia
Livello sporcizia	Identifica lo stato di sporcizia di ogni locale attraverso l'uso di sensori.		Locale
Parametri locale	Indicano grazie a dei sensori, la temperatura e l'umidità di ogni locale.		Locale
Riproduzione	Si selezionano i due animali da far riprodurre ai fini di ottenere un'elevata qualità del prodotto.		Animale, veterinario
Veterinario	Supervisiona la riproduzione degli animali.		Riproduzione, Gestazioni, Terapia, Esame
Gestazione	Quando la riproduzione ha avuto esito positivo.		Veterinario, Esame, Animale
Esame	Un esame è prescritto se la gestazione ha esito negativo.	Analisi, osservazione	Gestazione, Terapia, Animale, Veterinario
Farmaco	Farmaco da assumere durante le terapie.	Integratore	Terapia
Gravidanza	Animale che è in gravidanza.		Animale, Complicanza,
Complicanza	Si può verificare durante la gravidanza.	Aggravamento	Animale, Gravidanza, Visita
Zona	Determinata area delimitata dell'agriturismo.	Area	Attività di Pascolo, Mungitrice

Attività di pascolo	Avviene in una zona e in una fascia oraria, coinvolgendo tutti gli animali di un locale.		Locale, Zona, Posizione (GPS), Orario di rientro
Posizione (GPS)	Identifica la posizione dell'animale durante il pascolo.		Animale, Attività di pascolo
Orario di rientro	Orario di rientro alla fine del pascolo di ogni animale		Animale, Attività di pascolo

2.2Area Healthcare

L'area healthcare si occupa della salute e del benessere di ogni animale.

Termine	Descrizione	Sinonimi	Collegamenti
Scheda medica	Informazioni riguardo allo stato di salute di ogni animale.	Cartella	Animale, Indicatori
Indicatori	Rilevano lo stato di salute del paziente attraverso varie misurazioni. Possono essere soggettivi ed oggettivi.	Rilevatori, sensori	Animale, Scheda medica
Patologia	Malattia di cui è affetto un animale		Animale, Terapia
Terapia	Terapia prescritta dal veterinario da somministrare all'animale		Esame, Animale, Veterinario, Farmaco, Patologia
Visita	Visita di un animale effettuata da un veterinario	Controllo	Animale, Complicanza

2.3 Area Produzione

Nell'area produzione vengono gestiti i processi di produzione dei prodotti caseari e i dati attinenti alle mungiture.

Termine	Descrizione	Sinonimi	Collegamenti
Mungitrice	Macchinario usato dagli animali		Zona, Mungitura
Mungitura	Coinvolge animali di uno stesso locale		Animale, Mungitrice, Locale, Silos
Silos	Silos di raccolta latte	Magazzino, Deposito	Mungitura, Lotto
Formaggio	Prodotto caseario		Ricetta, Lotto
Ricetta	Specifica le fasi di produzione del formaggio		Formaggio, Cantina
Dipendenti Lotto	Tutti i dipendenti che lavorano in un lotto	Lavoratori	Lotto
Lotto	Identifica i prodotti caseari		Formaggio, Silos, Cantina, Magazzino, Laboratorio, Dipendenti Lotto
Laboratorio	Luogo in cui vengono lavorati i prodotti		Lotto
Cantina	Nella cantina vengono collocati i lotti che devono iniziare il processo di stagionatura		Lotto, Scaffale, Ricetta
Magazzino	Nel magazzino vengono collocati i prodotti che non necessitano di stagionatura		Lotto, Scaffale
Scaffale	Posizione specifica in cui viene riposto il prodotto		Magazzino, Cantina, Stagionatura
Stagionatura	Indica i prodotti che necessitano di stagionatura		Scaffale

2.4 Area Soggiorno

Grazie a tale area di ogni agriturismo, è possibile affittare delle camere ai clienti che desiderano trascorrere dei giorni di vacanza in luoghi affascinanti.

Termine	Descrizione	Sinonimi	Collegamenti
Prenotazione stanza	Prenotazione dell'utente (registrato e non).		Account, Tipo stanza, Prenotazione servizio, Pagamento, Utenti non registrati
Utenti non registrati	Coloro che prenotano una stanza senza avere un account		Prenotazione stanza, Pagamento, Prenotazione Escursione
Tipo stanza	Tipo di stanza dell'agriturismo (semplice o suite).		Prenotazione stanza
Servizio aggiuntivo	Servizi aggiuntivi solo per utenti che hanno prenotato una suite. Per tutti o alcuni giorni di permanenza.		Prenotazione servizio
Prenotazione servizio	Prenotazione dei servizi aggiunti (solo per suite). Ognuno ha una durata diversa.		Prenotazione stanza, Servizio aggiuntivo
Pagamento	Pagamento del soggiorno (sia per utenti registrati che non).	Compenso	Prenotazione stanza, Utenti non registrati
Escursione	Escursioni offerte dall'agriturismo.	Visita, Gita	Prenotazione escursione, Guida, Itinerario Escursione
Itinerario Escursione	Itinerario composto da un insieme di tappe.	Percorso	Escursione, Tappa
Tappa	Una tappa di un itinerario.	Fermata, sosta	Itinerario Escursione, Area
Area	Area dell'agriturismo che può essere	Luogo	Tappa

	presente in una tappa.		
Guida	Guida relativa all'escursione	Accompagnatore	Escursione
Prenotazione escursione	Prenotazione dell'escursione		Account, Utenti non registrati, Escursione

2.5 Area store

L'area store ha il compito di gestire le vendite online dei prodotti dell'agriturismo

Termine	Descrizione	Sinonimi	Collegamenti
Registrazione	Registrazione di un nuovo utente nel sito web.	Iscrizione	Account, Documento
Documento	Documento e specifiche.	ID	Registrazione
Account	Account creato con la registrazione.		Registrazione
Magazzino	Insieme di prodotti che possono essere venduti.		Prodotto
Prodotto	Prodotti presenti nello store online.	Elemento, articolo	Magazzino, Carrello, Reso, Recensione cliente
Ordine	Ordine del cliente (account).		Account, Carrello, Spedizione, Reso, Recensione cliente
Carrello	Insieme di prodotti di un determinato ordine.		Ordine, Prodotto
Spedizione	Spedizione dei prodotti.		Ordine, Itinerario Spedizione
Itinerario Spedizione	Tiene traccia della spedizione.		Spedizione, Centro Smistamento
Centro smistamento	Hub itinerario spedizione		Itinerario Spedizione
Reso	Reso del/i prodotto/i riferiti all'ordine.		Ordine, Prodotto
Recensione cliente	Recensione del cliente del relativo prodotto acquistato		Ordine, Prodotto

3. Dizionario dei dati

3.1 Area Allevamento

Entità	Attributi	Identificatore
Animale	Famiglia, DataNascita, Sesso, Razza, Altezza, Peso, DataAcquisto, DataArrivo	Codice
GPS	Longitudine, Latitudine, Time	ID
Specie		Nome
Locale	PtCardinaleFinestra, TipoPavimento, Altezza, Larghezza, Lunghezza	Codice
Stalla		Codice
Agriturismo	Nome, Citta, Indirizzo	Codice
Zona		Nome + Agriturismo
Pascolo	TimeInizio, TimeFine	Codice
Riproduzione	Time, Stato	Codice
SchedaDiGestazione		Riproduzione
InterventoDiControllo	Stato, DataControllo, DataProgrammata, Esito	Codice
Veterinario	Nome, Cognome, Parcella	Matricola
Esame	Time, DescrizioneProcedimento, Nome	Codice
Macchinario		Codice
Fornitore	Nome, Indirizzo, RagioneSociale	PartitaIVA
Allestimento		Codice
Mangiatoia		Codice
Abbeveratoio		Codice
Condizionamento Illuminazione	Tipo	Codice
StatoMangiatoia	Quantita	Istante + Mangiatoia
Foraggio	Peso, KCal, Frutta, Cereali, Piante, Proteine, Glucidi, Fibre, Tipo	Codice
StatoAbbeveratoio	Sali Minerali, Vitamine, Quantita	Istante + Abbeveratoio
Sensore	Temperatura, Umidita, LivSporco, LivMetano, LivAzoto	Codice
StatoSensore	Stato	Time + Sensore
Dipendente	Nome, Cognome, DataNascita, Tipologia	Codice

Gruppi		Codice
Recinzione	X1, X2, Y1, Y2	ID

Relazione	Entità Coinvolte	Attributi
Localizzato	Animale (1,1) GPS (0,1)	
Parentela	Animale (Figlio) (0,N) Animale (Genitore) (0,N)	
Appartenenza	Animale (1,1) Specie (0,N)	
Risiedere	Animale (1,1) Locale (1,N)	
Situata	Specie (0,N) Locale (1,1)	NumMassimo
Suddivisione	Locale (1,1) Stalla (1,N)	
Divisione	Agriturismo (1,N) Zona (1,1)	
Formazione	Zona (0,N) Pascolo (1,1)	
Andare	Animale (0,N) Pascolo (1,N)	
Effettua	Animale (0,N) Riproduzione (1,N)	
Controllare	Riproduzione (0,1) SchedaDiGestazione (1,1)	
Descrive	SchedaDiGestazione (1,N) InterventoDiControllo (1,1)	
Esegue	InterventoDiControllo (1,1) Veterinario (0,N)	
Monitorare	StatoSensore (1,1) Sensore (1,N)	
Assegnamento	Veterinario (0,N) SchedaDiGestazione (1,1)	
Supervisiona	Veterinario (0,N) Riproduzione (1,1)	
Seguire	Esame (1,1) InterventoDiControllo (0,1)	
Riferimento	InterventoDiControllo (0,1) Terapia (0,1)	
Vendita	Animale (1,1) Fornitore (0,N)	
Presenza	Locale (1,1) Allestimento (1,1)	
Suddivisa	Dipendente (0,1) Gruppi (1,N)	
Composizione1	Allestimento (1,N) Mangiatoia (1,1)	
Composizione2	Allestimento (1,N)	

	Abbeveratoio (1,1)	
Composizione3	Allestimento (1,N) Condizionamentolluminazione (1,1)	
Possedere1	Mangiatoia (1,N) StatoMangiaotia (1,1)	
Riporta	StatoMangiatoia (1,1) Foraggio (1,N)	
Locazione	Stalla (1,1) Agriturismo (1,N)	
Associare	Esame (0,1) Macchinario (1,1)	
Possedere2	Abbeveratoio (1,N) StatoAbbeveratoio (1,1)	
Sensorizzazione	Locale (1,1) Sensore (1,1)	
Delimitare	Zona (1,N) Recinzione (1,1)	
Lavorare	Dipendente (1,1) Agriturismo (1,N)	
Corrisponde	Locale (1,N) Pascolo (1,1)	

3.2Area Healthcare

Entità	Attributi	Identificatore
Terapia	DataInizioTerapia, Durata, Posologia	Codice
SchedaTerapia	DataFine	DataInizio + Terapia + Farmaco
Orario		Ora
Farmaco	PrincipioAttivo	NomeCommerciale + Patologia
Visita		Data + Animale + Veterinario
Quarantena		Time + Animale
SchedaMedica		Animale + Time
IndicatoriSoggettivi	Entita, Pt.Corpo, Tipo, NomeDisturbo	Codice
Patologia		NomePatologia
Indice	Nome	Codice

Relazione	Entità Coinvolte	Attributi
Prevede	Terapia (1,N) SchedaTerapia (1,1)	
Dosaggio	SchedaTerapia (0,N) Orario (1,N)	Dose
Utilizzare	Scheda Terapia (1,1) Farmaco (1,N)	
Riferimento	Terapia (1,1) Visita (0,N)	
Prescrive	Terapia (1,1) Veterinario (0,N)	
Occupare	Visita (1,1) Veterinario (0,N)	
Sottoporre	Visita (1,1) Animale (0,N)	
Inserito	Animale (0,N) Quarantena (1,1)	
Dotazione	Animale (1,N) SchedaMedica (1,1)	
Possibilita	SchedaMedica (0,N) IndicatoriSoggettivi (0,N)	
Basare	Terapia (1,1) Patologia (1,N)	
Attribuire	Patologia (1,N) Farmaco (1,1)	

Valutare	SchedaMedica(0,N) Indice (1,N)	Valore
----------	-----------------------------------	--------

3.3 Area Produzione

Entità	Attributi	Identificatore
Latte	Quantita, pH, Densita, Temperatura, Glucosio, Caseina, Lattoalbumina	Mungitura + Animale
Mungitura		Time + Mungitrice
Mungitrice	Modello, Marca	Codice
StatoSilos	Capacita, LivRiempimento, pH, Densita, Temperatura, Glucosio, Caseina, Lattoalbumina	Codice
Silos	Marca	Codice
Lotto	Data, Durata, DataScadenza	Codice
FaseLotto	Durata, Temperatura	NumFase + Lotto
Parametro		Nome
Laboratorio	Tipo	Codice
UnitaProdotto	Peso	Codice
ProdottoCaseario	ZonaOrigine, Tipo, GradoDeperibilita, Costo	Nome
Ricetta	Descrizione, DataStagionatura	ProdottoCaseario
Fasi	Durata	IdFase + Ricetta
ParametroIdeale		Nome
ScaffaleCantina	CapienzaMax	Codice
ScaffaleMagazzino	CapienzaMax	Codice
Cantina		Codice
Magazzino		Codice
StatoCantina	Temperatura, Umidita	Time, Cantina

Relazione	Entità Coinvolte	Attributi
Produzione	Animale (0,N) Latte (1,1)	
Posizione	GPS (0,1) Mungitrice (1,N)	
Riferimento	Latte (1,1) Mungitura (1,N)	
Utilizzare	Mungitura (1,1) Mungitrice (0,N)	
Immagazzinare	Latte (1,1) StatoSilos (0,N)	
Prelievo	StatoSilos (0,N) Lotto (1,N)	Quantita

Descritto	Lotto (1,N) FaseLotto (1,1)	
Avere	FaseLotto (1,N) Parametro (0,N)	Quantita
Produzione	Lotto (1,1) Laboratorio (0,N)	
Impiegato	Lotto (1,N) Dipendente (0,N)	
Lavorato	UnitaProdotto (1,1) Lotto (1,N)	
Presenza	UnitaProdotto (1,1) ProdottoCaseario (0,N)	
Caratterizzato	ProdottoCaseario (1,1) Ricetta(1,1)	
Suddivisa	Ricetta (1,N) Fasi (1,1)	
Trovare	Laboratorio (1,1) Agriturismo (1,N)	
Stagionare	UnitaProdotto (0,1) ScaffaleCantina (0,N)	DataInizio
Posizionato	UnitaProdotto (0,1) ScaffaleMagazzino (0,N)	
Situato1	ScaffaleCantina (1,1) Cantina (1,N)	
Situato2	ScaffaleMagazzino (1,1) Magazzino (1,N)	
Monitoraggio	Cantina(1,N) StatoCantina (1,1)	
Appartiene1	Cantina (1,1) Agriturismo (1,N)	
Appartiene2	Magazzino (1,1) Agriturismo (1,N)	
Composta	Fase (1,N) ParametroIdeale (1,N)	Quantita
Possiede	StatoSilos (1,1) Silos (1,1)	

3.4 Area Soggiorno

Entità	Attributi	Identificatore
Area	DurataSosta	Ora + Nome + Itinerario
Itinerario		Codice
Escursione	Costo, OraFine, OraInizio, GiornoSettimana	Nome
Prenotazione	Anticipo, DataArrivo, DataPartenza	Id
ServizioAggiuntivo		TimeInizio, TimeFine, Prenotazione, Servizio
Servizio	CostoUnitario	Nome
Pagamento	Importo	Time, Prenotazione
Stanza	CostoStanza, Tipo	NumStanza
Letto	TipoLetto	CodLetto
Registrato	Nome, Cognome, Telefono, CodCarta, Tipologia, NumDocumento, Scadenza, Ente	Codice
NonRegistrato	CodiceCarta	Codice

Relazione	Entità Coinvolte	Attributi
Guidata	Dipendente (0,N) Escursione (1,1)	
Posizione	Agriturismo (1,N) Area (1,1)	
Formato	Area (1,N) Itinerario (1,N)	
Prevedere	Itinerario (1,N) Escursione (1,1)	
Associare	Escursione (0,N) Prenotazione (0,N)	DataPrenotazione
Richiedere	Prenotazione (0,N) ServizioAggiuntivo (1,1)	
Oggetto	ServizioAggiuntivo (1,1) Servizio (0,N)	
Terminare	Prenotazione (1,1) Pagamento (1,1)	
Riferimento	Prenotazione (0,N) Stanza (0,N)	
Trovare	Stanza (1,N) Letto (1,1)	
Effettua1	Prenotazione (0,1) Registrato (0,N)	
Effettua2	Prenotazione (0,1) NonRegistrato (0,N)	

3.5 Area Store

Entità	Attributi	Identificatore
Account	Password, DataIscrizione, Domanda, Risposta	NomeUtente
Ordine	Stato, Time	Codice
Recensione	Testo, Gradimento, Qualita, Gusto, Conservazione, Data	Codice
Reso	DataReso	Codice
Spedizione	Stato	Codice
ParametriQualita		Nome
Hub	Citta, Indirizzo	Codice

Relazione	Entità Coinvolte	Attributi
Corrisponde	Registrato (1,1) Account (1,1)	
PuòEffettuare	Account (0,N) Ordine (1,1)	
Composizione	Ordine (0,N) UnitaProdotto (0,1)	
Selezione	Ordine (0,N) ProdottoCaseario (0,N)	
Valutazione	UnitaProdotto (0,1) Recensione (1,1)	
Possibilita	UnitaProdotto (0,1) Reso (1,1)	
Controllo	Reso (0,N) ParametriQualita (0,N)	Valutazione
Trasporto	UnitaProdotto (0,N) Spedizione (1,N)	DataPrevista, DataEffettiva
Prevedere	Spedizione (1,N) Hub (1,N)	Time

4. Attributi derivati

Entità: Foraggio

▪ **Attributo:** *Proteine, Glucidi, Fibre*

Dall'attributo *Peso* si ricava il suo peso in grammi dividendolo per 1000 (P_G).

Successivamente si calcola il peso (in grammi) di ognuna delle componenti del foraggio (piante, cereali e frutta) grazie agli attributi *Piante, Cereali, Frutta*.

Dunque si ricavano P_p , P_c e P_f , che rispettivamente indicano il peso delle piante, dei cereali e della frutta:

$$P_p = \frac{P_G \times PercentualePiante}{100};$$

$$P_c = \frac{P_G \times PercentualeCereali}{100};$$

$$P_f = \frac{P_G \times PercentualeFrutta}{100}.$$

Nella tabella sottostante è riportata la quantità di proteine, glucidi e fibre per cereali, piante e frutta. Ogni dato numerico è stato calcolato facendo la media fra circa 20 elementi per tipologia. I dati sono in grammi.

	Proteine	Glucidi	Fibre
Cereali	11.5	75	2.5
Piante	2.7	3.2	3
Frutta	1.2	16	2.8

Successivamente si possono ricavare le quantità di proteine, glucidi e fibre dell'intero foraggio, che sono inserite rispettivamente negli attributi *Proteine, Glucidi, Fibre*.

Gli attributi *Proteine, Glucidi, Fibre* saranno dati da:

$$\sum_{i=1}^N P_i \times DatoTabella_i,$$

dove N sarà uguale a 3; $i = \{\text{Cereali, Piante, Frutta}\}$ e P_i rappresenta il peso dell'elemento i e $DatoTabella_{i,j}$ il corrispettivo dato. L'operazione di somma sarà effettuata tre volte, rispettivamente per *Proteine, Glucidi, Fibre*. Il loro valore è in grammi.

▪ **Attributo:** *Kcal*

È noto che:

1 g di **proteine** corrisponde a **4.0** Kcal;

1 g di **glucidi** corrisponde a **3.8** Kcal;

1 g di **fibre** corrisponde a **2.0** Kcal.

Da ciò si può calcolare l'attributo *Kcal* (il valore è in Kcal):

$$Kcal = \frac{\sum_{i=0}^N Quantita_i \times Calorie_i}{1000}$$

Dove $N = 3$, $i = \{\text{proteine, glucidi, fibre}\}$, $Quantita_i$ rappresenta l'attributo *quantità* del corrispettivo i , $Calorie_i$ ha lo stesso ragionamento di quest'ultimo.

Entità: Pagamento

- **Attributo:** *Importo*

$$Importo = \sum_{j=1}^m \left(CostoStanza_j \times (DataPartenza_j - DataArrivo_j) \right) + \sum_{k=1}^p Costo_k + \sum_{i=1}^n (CostoUnitario_i \times (TimeFine_i - TimeInizio_i)) - Anticipo$$

Dove:

- *CostoStanza* è in **Stanza** e rappresenta il costo giornaliero della camera;
- *DataArrivo* e *DataPartenza* sono in **Prenotazione** e rappresentano la durata del soggiorno;
- *Costo* è in **Escursione** e rappresenta il costo di un'escursione;
- *CostoUnitario* è in **Servizio** e rappresenta il costo orario del servizio;
- *TimeInizio* e *TimeFine* sono in **ServizioAggiuntivo** e rappresentano la durata del servizio aggiuntivo;
- *Anticipo* è in **Prenotazione** e rappresenta la quota versata dall'utente in fase di prenotazione.

- **Relazione:** *Situata*

- **Attributo:** *NumMassimo*

Ipottizziamo delle dimensioni standard per ogni specie:

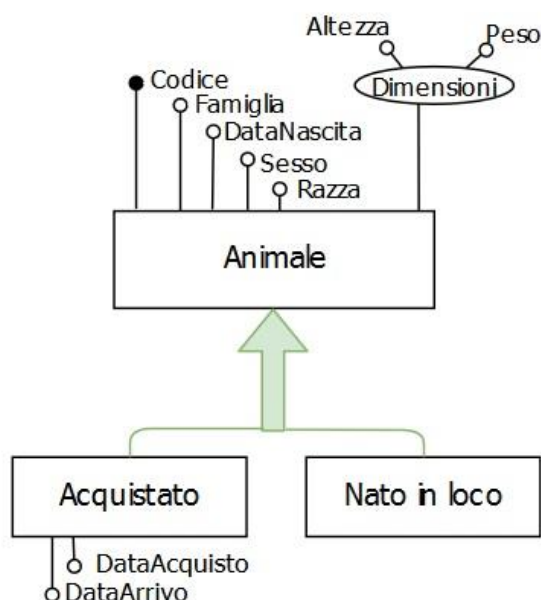
- a) **Ovini** → Lunghezza: 140 cm, Larghezza: 35 cm
- b) **Bovini** → Lunghezza: 180 cm, Larghezza: 45 cm
- c) **Caprini** → Lunghezza: 85 cm, Larghezza: 30 cm

$$NumMassimo = \frac{Larghezza \times Lunghezza}{LarghezzaSpecie \times LunghezzaSpecie}$$

- **Entità:** StatoSilos
 - **Attributo:** LivRiempimento
Ogni qual volta che si aggiunge al silos del latte dall'entità Latte verrà incrementato, e ogni volta che si svuota da parte dell'entità Lotto si decrementa.

5. Generalizzazioni

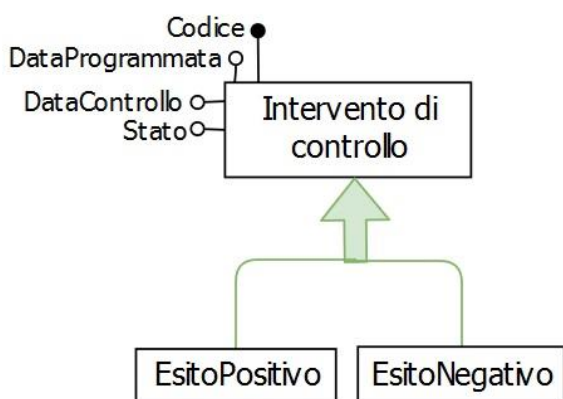
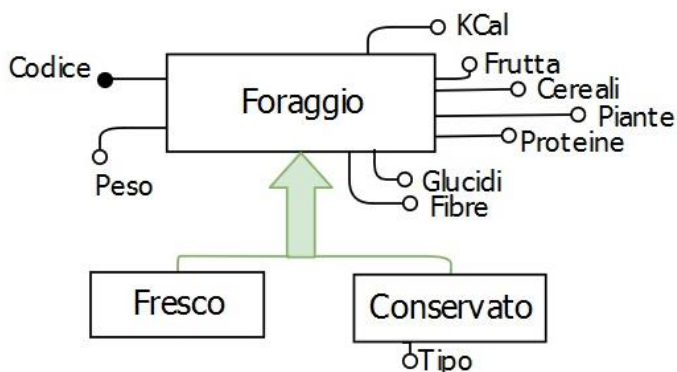
Diverse entità presenti nello schema sono state rappresentate come generalizzazioni. Sono state fatte queste scelte di progettazione, in quanto i legami tra le entità risultavano molto simili, sfruttando quindi, l'ereditarietà tra entità padre/figlie, ottenendo dei vantaggi in termini di attributi e soprattutto evitando ridondanze.



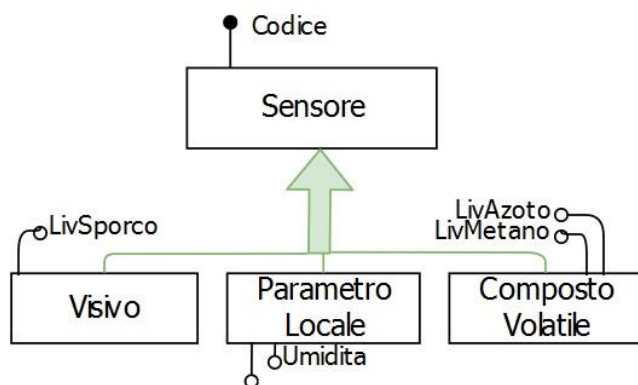
Gli animali degli agriturismi possono essere nati in loco o acquistati da fornitori, in questo caso occorre specificare sia la data di acquisto che la data di arrivo in agriturismo. La generalizzazione è **totale** poiché ogni animale deve essere obbligatoriamente o acquistato o autoctono, ed è **esclusiva** in quanto se un animale viene acquistato non può essere autoctono e viceversa.

Il foraggio può essere dato agli animali, fresco o conservato, se conservato occorre specificare il tipo ovvero se è insilato o sotto forma di fieno.

La generalizzazione è **totale** poiché le tipologie di foraggio sono solo quelle due, ed è **esclusiva**.

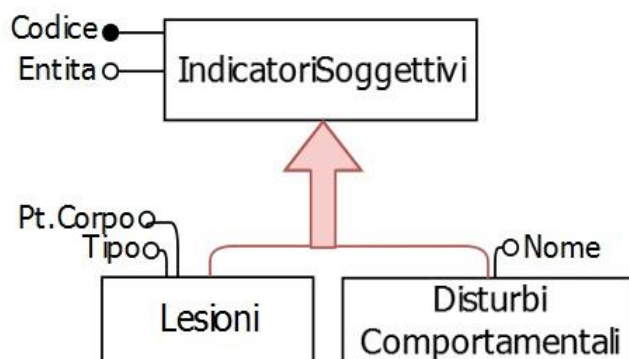
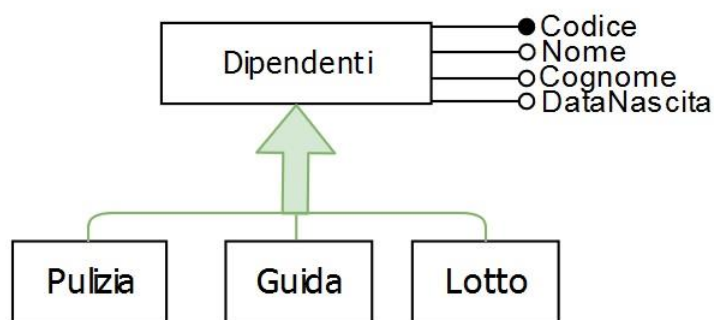


Ogni intervento di controllo può avere un esito, positivo o negativo, di conseguenza la generalizzazione è **totale** ed **esclusiva** in quanto un esito esclude il suo opposto.



Ogni locale è dotato di sensori che garantiscono l'efficienza della manutenzione dell'agriturismo. Per ogni sensore si definisce il proprio stato. Questo tipo di generalizzazione è **totale** in quanto ci possono essere altri tipi di sensori oltre a quelli indicati, e **non esclusiva** poiché in un locale possono esserci diversi tipi di sensori.

Tutti i dipendenti dell'agriturismo sono stati divisi in base alla categoria a cui appartengono. Questo tipo di generalizzazione è **totale ed esclusiva** in quanto un dipendente può essere appartenere solo ad uno dei figli della generalizzazione.



Gli indicatori soggettivi permettono di valutare lo stato di salute di ogni animale valutando le lesioni o i disturbi comportamentali.

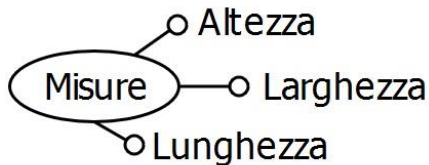
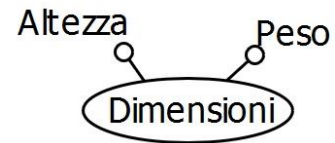
La generalizzazione è **totale**, in quanto quelli rappresentati sono gli unici indicatori soggettivi, e **non esclusiva** in quanto uno stesso animale può presentare entrambi i sintomi.

I clienti dell'agriturismo sono stati divisi in due gruppi "registrato" e "non registrato". La generalizzazione è **totale**, in quanto ogni cliente che pernotta, acquista o usufruisce dei servizi dell'agriturismo può essere registrato o meno, di conseguenza risulta essere anche **esclusiva** dato che un'entità esclude l'altra.



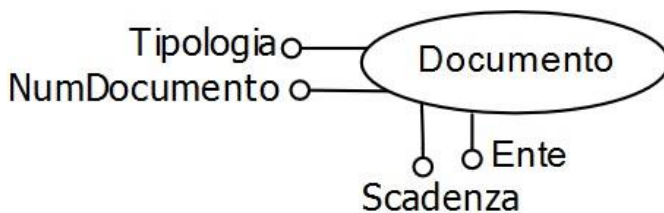
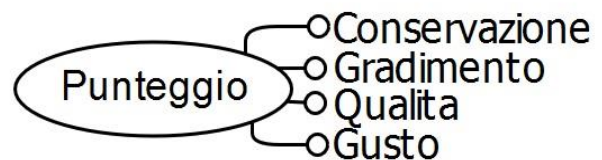
6. Attributi composti

Per indicare le dimensioni di ogni animale, è stato creato l'attributo composto dimensioni che comprende l'altezza e il peso di ogni singola specie presente nell'agriturismo.



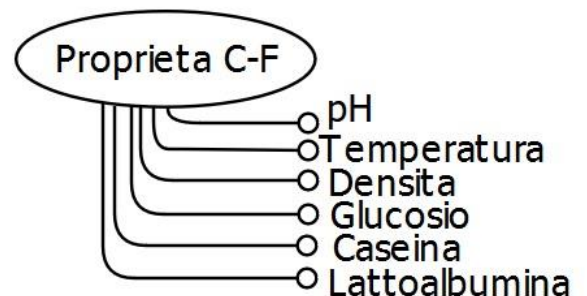
Ogni locale ha le proprie misure. Questo attributo composto è importante perché grazie ai parametri che contiene (Altezza, Lunghezza, Larghezza) è possibile adattare le specie ai locali.

Questo attributo composto, si riferisce alle recensioni. Ogni recensione possiede dei "punteggi", ovvero dei parametri che permettono di valutare se l'utente è stato soddisfatto o meno del prodotto da lui acquistato. I parametri da specificare sono: Conservazione, Gradimento, Qualità e Gusto.



Tutti i clienti registrati, devono specificare un documento in corso di validità. Si è deciso di utilizzare l'attributo composto "Documento" per raggruppare tutte le informazioni riguardanti il documento del cliente. Tra queste informazioni ci sono: la tipologia, il numero del documento, la scadenza e l'ente che lo ha rilasciato.

Tutte le proprietà chimico-fisiche riguardante il latte contenuto nei silos, sono state raggruppate nell'attributo composto "Proprieta C-F". Queste proprietà comprendono il pH, la temperatura, la densità, il glucosio, la caseina e la lattoalbumina.

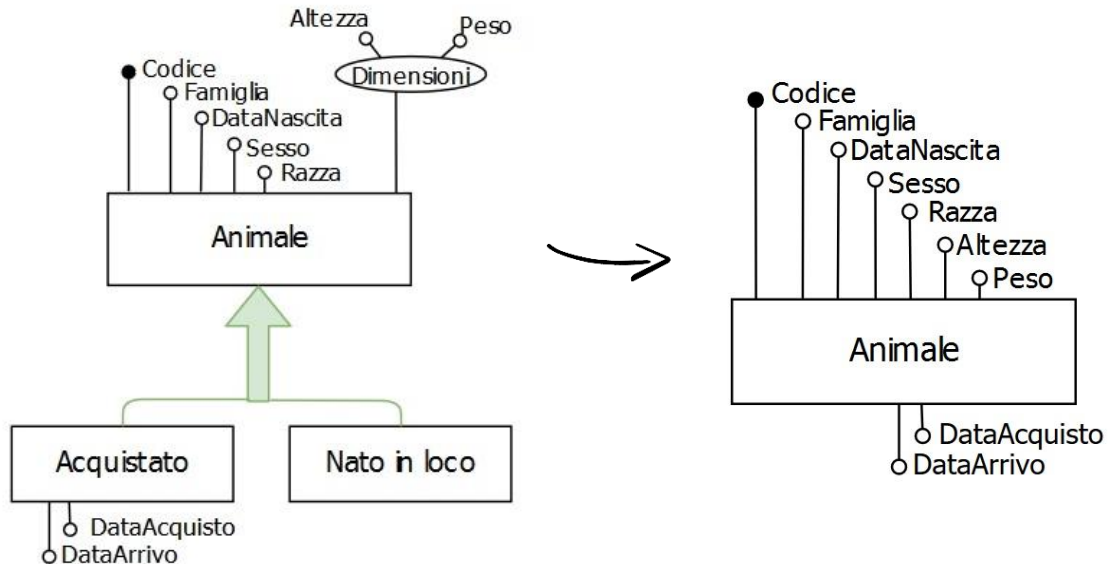


I confini delle zone dell'agriturismo sono dotati di recinzioni mobili che si adattano alla dimensione del pascolo corrente. Per fare questo, occorre salvare le posizioni dei due cardini principali che formano un lato. Per questo motivo, si è scelto di creare due attributi multipli, punto1 e punto2, i quali al loro interno hanno le relative coordinate



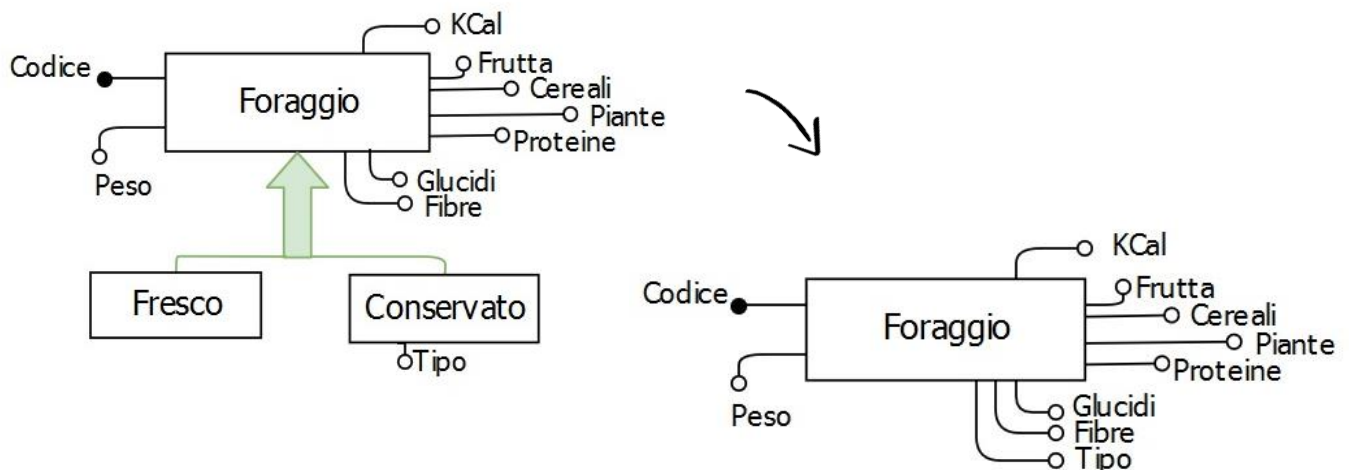
7. Traduzione delle generalizzazioni ed eliminazione attributi composti/multivalore

- Accorpamento dei figli sul padre:



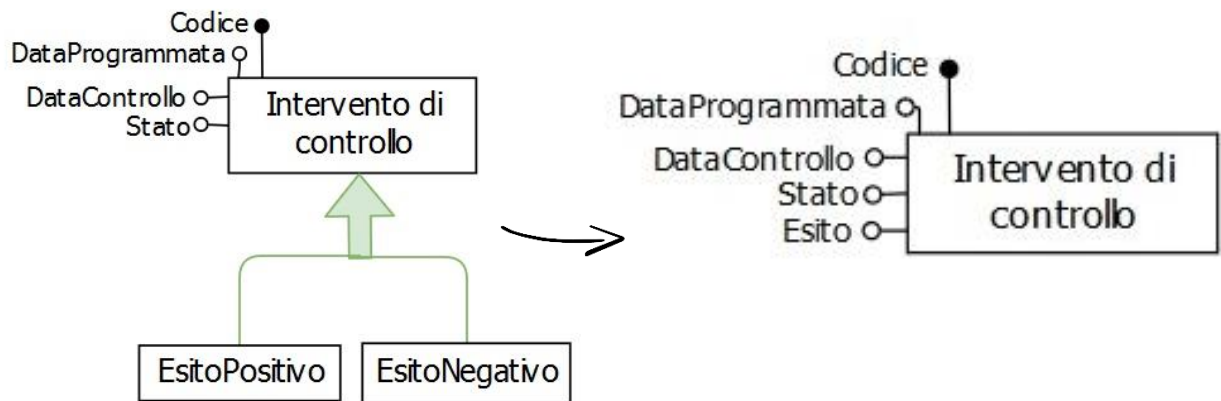
Si è deciso di tradurre questa generalizzazione con l'accorpamento dei figli sul padre in quanto entrambe le entità figlie non avevano nessuna relazione in comune con altre entità. Quindi sono state eliminate le due entità figlie e i loro attributi sono stati aggiunti nell'entità padre. Anche l'attributo composto è stato tradotto, collegando direttamente gli attributi Altezza e Peso all'entità Animale.

- Accorpamento dei figli sul padre:



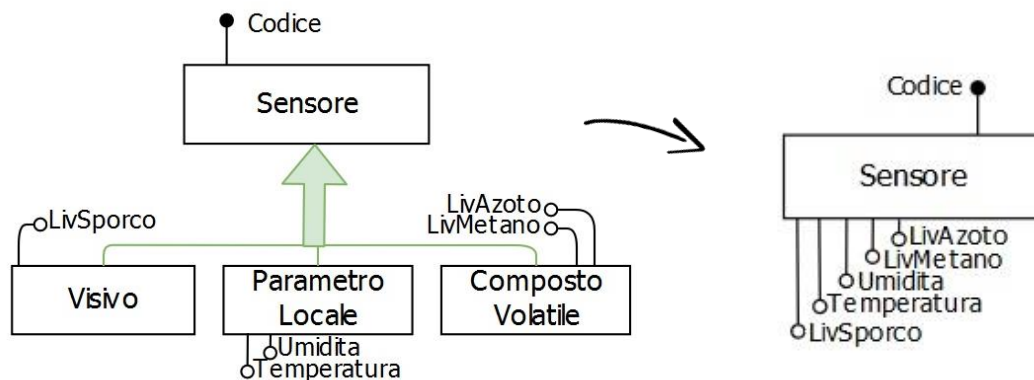
Questa generalizzazione è stata tradotta con l'accorpamento dei figli sul padre. Le due entità figlie sono state tolte e l'attributo Tipo è stato aggiunto direttamente all'entità padre ovvero Foraggio.

- Accorpamento dei figli sul padre:



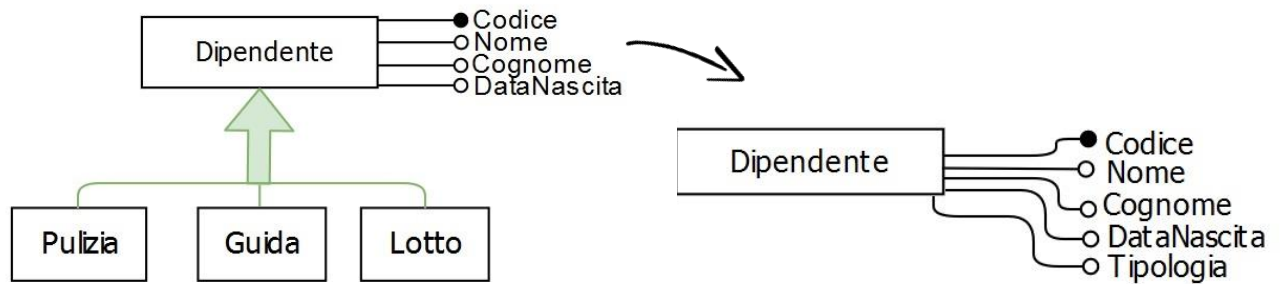
Anche in questo caso è stato applicato l'accorpamento dei figli sul padre. Quindi le entità figlie sono state rimosse e per classificare l'esito dell'intervento di controllo, abbiamo aggiunto un attributo "Esito" all'entità padre.

- Accorpamento dei figli sul padre:



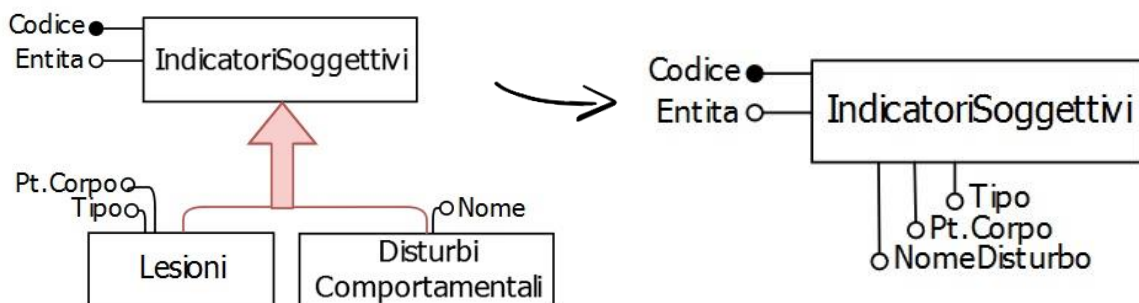
La generalizzazione "Sensore" è stata tradotta accorpendo i figli al padre, e quindi aggiungendo gli attributi di "Visivo", "Parametro Locale" e "Composto Volatile" all'entità padre "Sensore".

- Accorpamento dei figli sul padre:



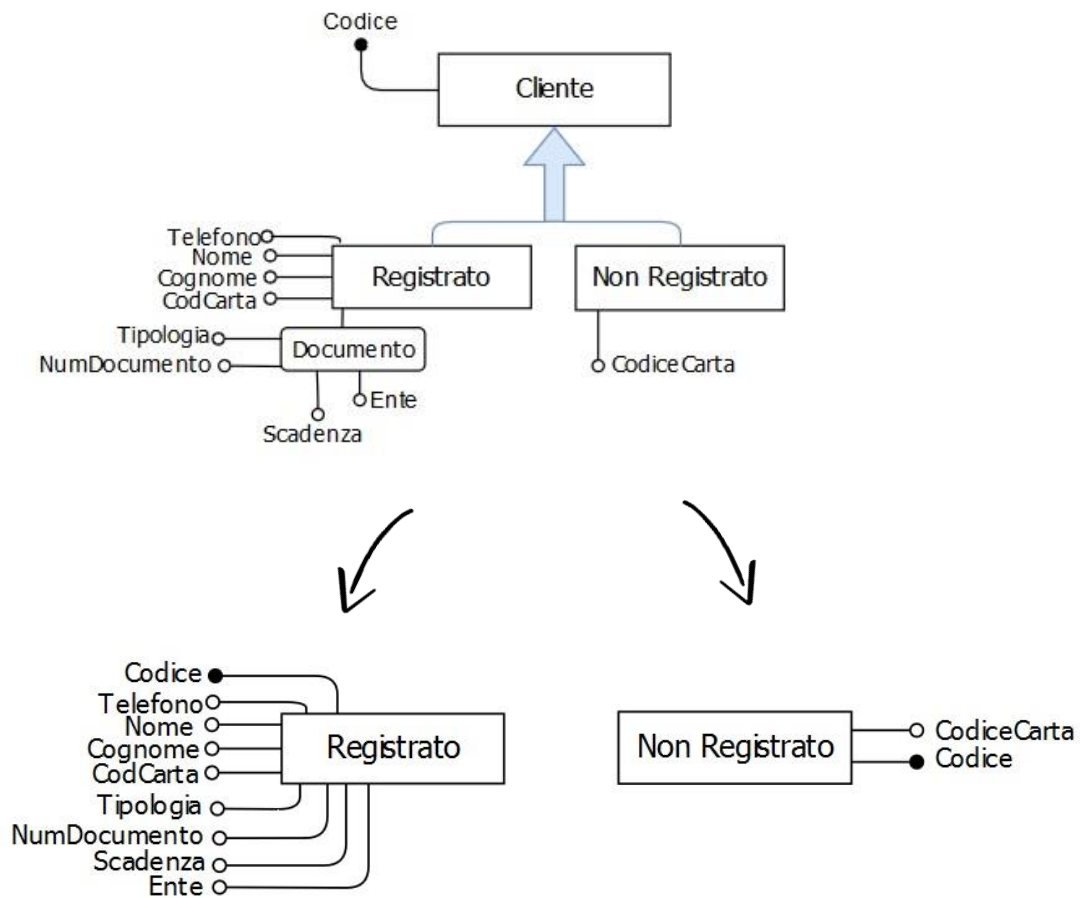
La generalizzazione “Dipendente” è stata tradotta accorpendo i figli al padre, per questo motivo sull’entità padre Dipendente è stato inserito un nuovo attributo “Tipologia” che specifica la categoria del dipendente, quindi se si occupa delle pulizie, se è una guida o se è impiegato nel lotto.

- Accorpamento dei figli sul padre:



La generalizzazione “IndicatoriSoggettivi” è stata tradotta accorpendo i figli al padre. Gli attributi delle due entità figlie sono stati ricopiati nell’entità padre, alla quale è stato aggiunto anche un ulteriore attributo, “Tipo”, che indica se si tratta di una lesione o di un disturbo comportamentale.

- Accorpamento del padre sui figli



Questa è l'unica generalizzazione che è stata tradotta, applicando l'accorpamento del padre sui figli. Per tale motivo è stato aggiunto l'attributo "codice" di Cliente alle due entità figlie.

8. Ristrutturazione del diagramma E-R

Il primo passo per la ristrutturazione del diagramma è la valutazione delle prestazioni, con l'obiettivo di valutare le ridondanze all'interno del diagramma ed eliminare sprechi di memoria ed un maggior numero di accessi necessari per mantenere la base di dati in uno stato coerente.

Per decidere se eliminare o meno una ridondanza si ricorre alla *Tavola dei volumi* e alla *Tavola degli accessi*.

8.1 Tavola dei Volumi

8.1.1 Tavola dei volumi area allevamento:

AREA ALLEVAMENTO		
Concetto	Tipo	Volume
Animale	E	500
Fornitore	E	20
Locale	E	25
Specie	E	3
Pascolo	E	$500 * 20 = 10000$
Zona	E	10
Stalla	E	5
Agriturismo	E	1
Dipendente	E	100
Sensore	E	25
StatoSensore	E	$25 * 20 = 500$
Allestimento	E	25
Mangiatoia	E	$25 * 5 = 125$
Abbeveratoio	E	$25 * 5 = 125$
CondizionamentoIlluminazione	E	$50 * 5 = 250$
StatoMangiatoia	E	$125 * 20 = 2500$
StatoAbbeveratoio	E	$125 * 20 = 2500$
GPS	E	10000
Foraggio	E	50
Gruppi	E	5
Riproduzione	E	250
SchedaDiGestazione	E	150

InterventoDiControllo	E	150 * 3 = 450
Recinzione	E	4 * 10 = 40
Esame	E	10
Macchinario	E	5
Parentela	R	200
Vendita	R	50
Effettua	R	200
Localizzato	R	5000 (1000/2)
Risiedere	R	500
Presenza	R	25
Composizione1	R	125
Composizione2	R	125
Composizione3	R	250
Possedere1	R	2500
Possedere2	R	2500
Riporta	R	2500
Monitorare	R	1500
Suddivisa	R	25
Lavorare	R	100
Sensorizzazione	R	25
Locazione	R	5
Suddivisione	R	25
Situata	R	25
Formazione	R	10000
Divisione	R	10
Appartenenza	R	500
Andare	R	10000
Supervisiona	R	250
Assegnamento	R	150
Controllare	R	150
Descrive	R	450

Esegue	R	450
Seguire	R	10
Riferimento	R	200
Associare	R	5
Delimitare	R	40
Corrisponde	R	10000

8.1.2 Tavola dei volumi area healthcare:

AREA HEALTHCARE		
Concetto	Concetto	Concetto
Veterinario	E	20
Visita	E	$500 * 10 = 5000$
Terapia	E	7500
Patologia	E	20
SchedaTerapia	E	$2 * 7500 = 15000$
Orario	E	3
Farmaco	E	60
SchedaMedica	E	5000 (= visita)
IndicatoriSoggettivi	E	3000
Indice	E	10
Quarantena	E	30
Indice	E	10
Dosaggio	R	$15000 * 1,5 = 22500$
Utilizzare	R	15000
Prevede	R	15000
Prescrive	R	7500
Occupare	R	5000
Riferimento	R	7500
Attribuire	R	60
Sottoporre	R	5000

Basare	R	7500
Inserito	R	30
Dotazione	R	5000
Possibilita	R	$5000 * 3 = 15000$
Valutare	R	$5000 * 10 = 50000$

8.1.3 Tavola dei volumi area produzione:

AREA PRODUZIONE		
Concetto	Tipo	Volume
Latte	E	$250 * 20 = 5000$
Mungitura	E	5000
Mungitrice	E	125
Silos	E	35
StatoSilos	E	35
ParametroIdeale	E	10
Fasi	E	5
Ricetta	E	40
ProdottoCaseario	E	40
UnitaProdotto	E	$40 * 100 = 4000$
ScaffaleCantina	E	50
ScaffaleMagazzino	E	100
Cantina	E	10
Magazzino	E	10
StatoCantina	E	$10 * 20 = 200$
Lotto	E	100
FaseLotto	E	5
Parametro	E	10
Laboratorio	E	5
Produzione	R	5000
Posizione	R	125
Riferimento	R	5000
Utilizzare	R	5000

Immagazzinare	R	5000
Possiede	R	35
Prelievo	R	$2 * 5000 = 10000$
Composta	R	$5 * 3 = 15$
Suddivisa	R	5
Caratterizzato	R	40
Presenza	R	4000
Posizionato	R	2000
Stagionare	R	2000
Situato1	R	50
Situato2	R	100
Appartiene1	R	10
Appartiene2	R	10
Monitoraggio	R	200
Trovare	R	5
Lavorato	R	4000
Produzione	R	$5000(1000/2)$
Impiegato	R	$5 * 100 = 600$
Descritto	R	5
Avere	R	40

8.1.4 Tavola dei volumi area soggiorno:

AREA SOGGIORNO		
Concetto	Tipo	Volume
Area	E	20
Itinerario	E	10
Escursione	E	50
Prenotazione	E	500
ServizioAggiuntivo	E	$50 * 30 = 150$
Servizio	E	12

Stanza	E	35
Letto	E	90
Pagamento	E	500
Registrato	E	200
NonRegistrato	E	50
Guidata	R	50
Posizione	R	20
Prevedere	R	50
Formato	R	20
Oggetto	R	150
Associare	R	200
Richiedere	R	150
Effettua1	R	200
Effettua2	R	100
Riferimento	R	300
Trovare	R	90
Terminare	R	500

8.1.5 Tavola dei volumi area store:

AREA STORE		
Concetto	Tipo	Volume
Account	E	200
Ordine	E	300
Spedizione	E	300
Hub	E	100
Recensione	E	250
Reso	E	40
ParametriQualita	E	5
Corrisponde	R	200
PuoEffettuare	R	300

Composizione	R	$300 * 3 = 900$
Selezione	R	$300 * 3 = 900$
Prevedere	R	$300 * 50 = 15000$
Trasporto	R	100
Possibilita	R	40
Controllo	R	$40 * 3 = 120$
Valutazione	R	250

8.2. Tavola delle Operazioni

TAVOLA DELLE OPERAZIONI			
Operazione	Descrizione	Tipo	Frequenza
1	Inserimento animale acquistato da un nuovo fornitore	I	2 al giorno
2	Visualizzare per ogni locale lo stato attuale dei loro sensori	B	10 al giorno
3	Visualizzare i veterinari che hanno effettuato interventi di controllo con esito negativo	B	1 a settimana
4	Visualizzare la data di nascita degli animali che hanno avuto disturbi comportamentali	B	2 a settimana
5	Inserimento Lotto	I	3 a settimana
6	Visualizzare servizi aggiuntivi per ogni prenotazione	B	5 a settimana
7	Leggere i dati di ogni stato silos (incluso il livello di riempimento)	I	2 a giorno
8	Stampare l'importo di ogni pagamento	B	5 al giorno

8.2.1 OPERAZIONE 1 - Inserimento animale acquistato da un nuovo fornitore

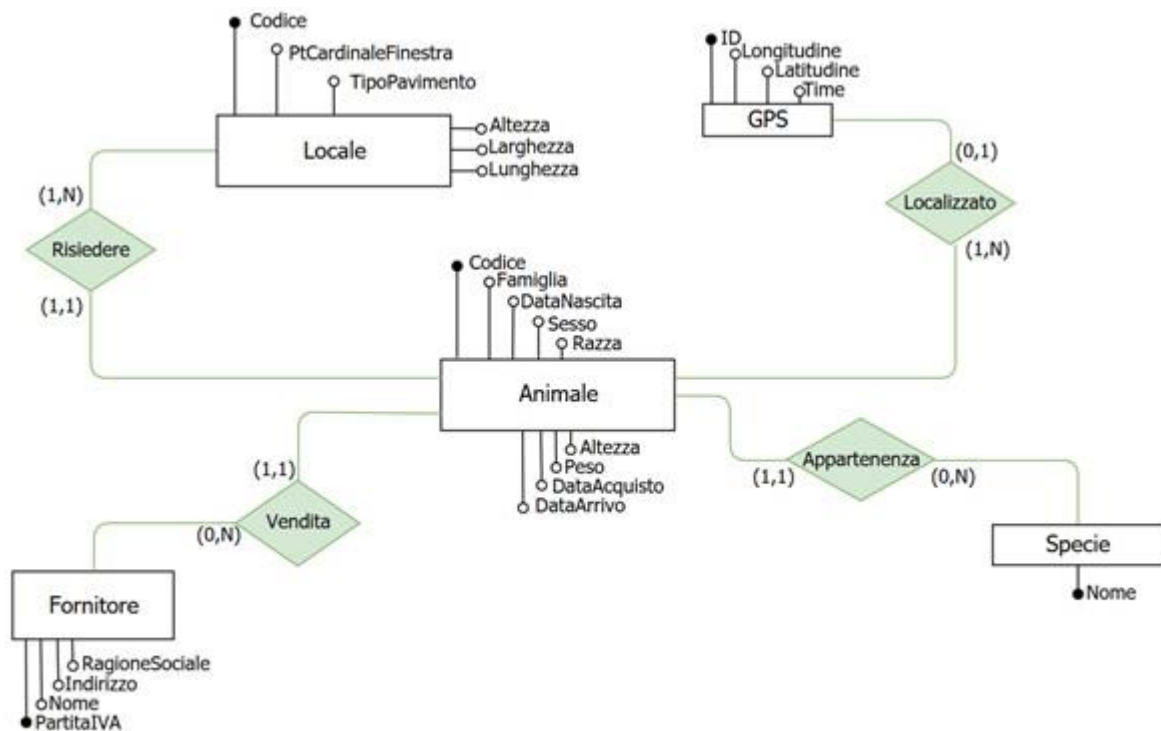


Tavola dei volumi		
Concetto	Tipo	Volume
Animale	E	500
Fornitore	E	20
Locale	E	25
GPS	E	625
Specie	E	3
Vendita	R	50
Risiedere	R	500
Localizzato	R	5000 (1000/2)
Appartenenza	R	500

Accessi	Tipo operazione	Tipo costruito	Nome costruito	Descrizione
1	S	E	Animale	Inserimento dati animale acquistato
1	S	R	Vendita	Inserimento nuova coppia fornitore-Animale

1	S	E	Fornitore	Inserimento record Fornitore
1	S	R	Risiedere	Inserimento nuova coppia Animale-Locale
1	L	E	Locale	Ricerca del locale d'interesse
1	S	R	Localizzato	Inserimento nuova coppia GPS-Animale
1	S	E	GPS	Inserimento del nuovo record riferito all'animale
1	S	R	Appartenenza	Inserimento nuova coppia Specie-Animale
1	L	E	Specie	Ricerca della specie d'interesse

Questa operazione viene effettuata 2 volte al giorno, di conseguenza si ottiene:

- Scrittura: $6 \times 2 = 12$ al giorno
- Lettura: $2 \times 2 = 4$ al giorno

8.2.2 OPERAZIONE 2 - Visualizzare per ogni locale lo stato attuale dei loro sensori

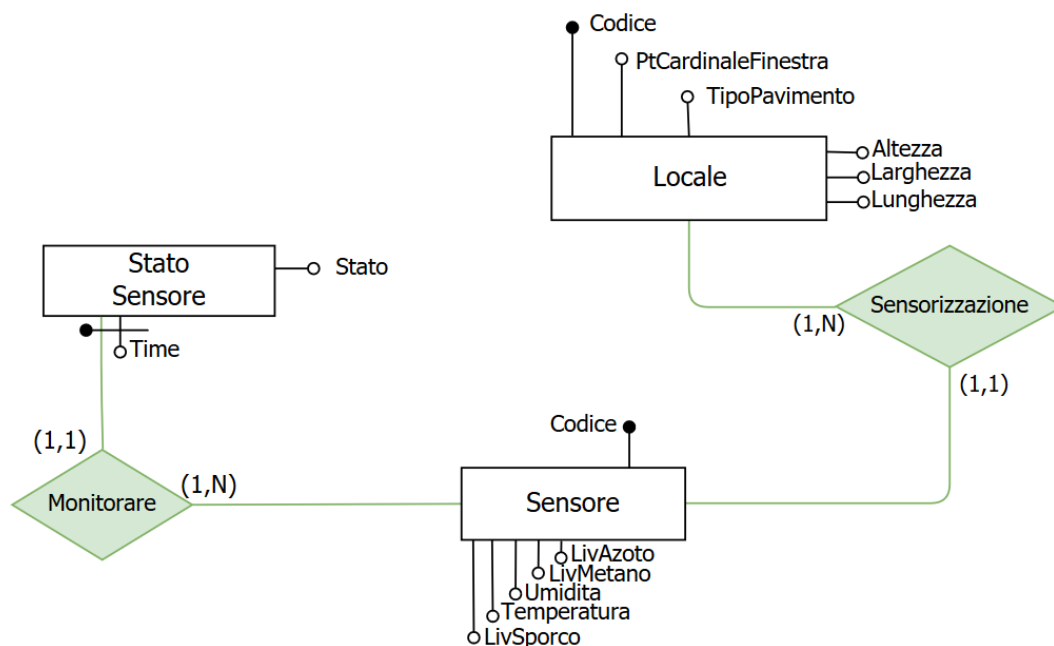


Tavola dei volumi		
Concetto	Tipo	Volume
Locale	E	25
Sensore	E	25
StatoSensore	E	500
Sensorizzazione	R	25
Monitorare	R	1500

Accessi	Tipo operazione	Tipo costruito	Nome costruito	Descrizione
25	L	E	Locale	Lettura dei record del Locale
25	L	R	Sensorizzazione	Lettura delle coppie Locale - Sensore per ogni locale
25	L	E	Sensore	Lettura dei sensori di ogni locale
1500	L	R	Monitorare	Lettura delle coppie sensore-StatoSensore per ogni locale
500	L	E	StatoSensore	Lettura dello stato del sensore per ogni sensore di ogni locale

Questa operazione viene effettuata 10 volte al giorno, di conseguenza si ottiene:

- Scrittura: 0
- Lettura: $10 \times [25 + 25 + 25 + 500 + 1500] = 20750$ al giorno

8.2.3 OPERAZIONE 3 - Visualizzare i veterinari che hanno effettuato interventi di controllo con esito negativo

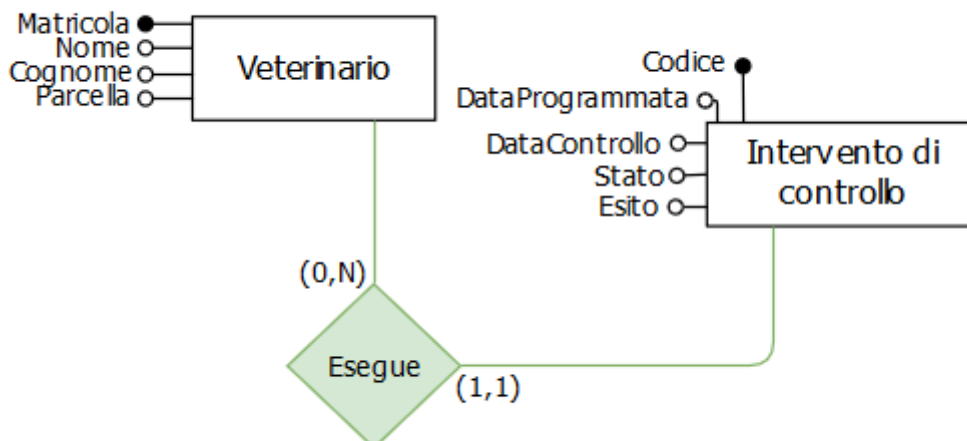


Tavola dei volumi		
Concetto	Tipo	Volume
Veterinario	E	20
InterventoDiControllo	E	450
Esegue	R	450

Accessi	Tipo operazione	Tipo costruito	Nome costruito	Descrizione
20	L	E	Veterinario	Lettura dei record dei Veterinari
450	L	R	Esegue	Lettura coppie Veterinario-InteventoDiControllo
450	L	E	InterventoDiControllo	Lettura InteventoDiControllo

Questa operazione viene effettuata 1 volta a settimana, di conseguenza si ottiene:

- Scrittura: 0
- Lettura: $1 \times [(450 \times 2) + 20] = 920$ a settimana

8.2.4 OPERAZIONE 4 - Visualizzare la data di nascita degli animali che hanno avuto disturbi comportamentali

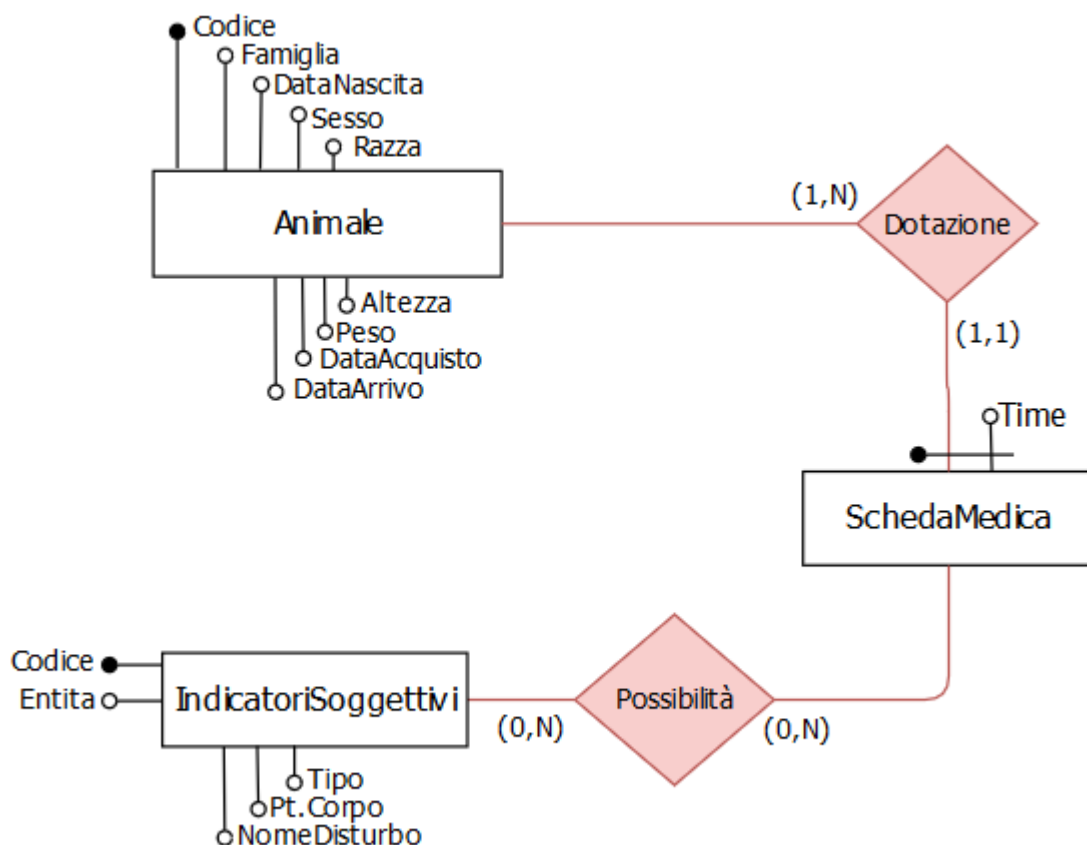


Tavola dei volumi		
Concetto	Tipo	Volume
Animale	E	500
SchedaMedica	E	5000
IndicatoriSoggettivi	E	3000
Dotazione	R	5000
Possibilità	R	15000

Accessi	Tipo operazione	Tipo costruito	Nome costruito	Descrizione
500	L	E	Animale	Lettura dei record degli animali
5000	L	R	Dotazione	Lettura coppie Animale-SchedaMedica
5000	L	E	SchedaMedica	Lettura dei record di SchedaMedica

15000	L	R	Possibilita	Lettura coppie SchedaMedica - IndicatoriSoggettivi
3000	L	E	IndicatoriSoggettivi	Lettura dei record di <u>IndicatoriSoggettivi</u>

Questa operazione viene effettuata 2 volte a settimana, di conseguenza si ottiene:

- Scrittura: 0
- Lettura: $2 \times [500 + 5000 + 5000 + 15000 + 3000] = 57000$ a settimana

8.2.5 OPERAZIONE 5 - Inserimento Lotto

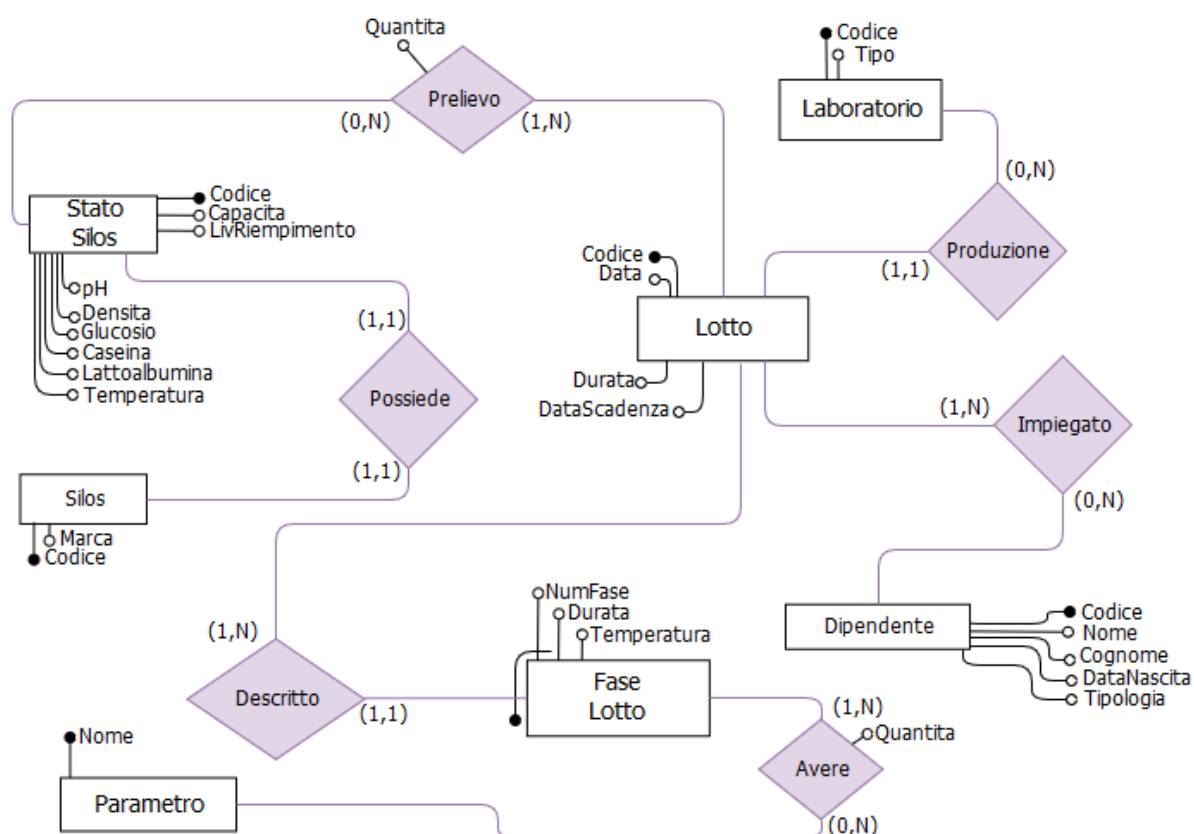


Tavola dei volumi		
Concetto	Tipo	Volume
Lotto	E	100
StatoSilos	E	35
FaseLotto	E	5
UnitaProdotto	E	4000
Prelievo	R	10000
Descritto	R	5
Avere	R	40
Produzione	R	100
Impiegato	R	500
Lavorato	R	4000

Accessi	Tipo operazione	Tipo costruito	Nome costruito	Descrizione
1	S	E	Lotto	Inserimento Lotto

1	L	R	Prelievo	Letture coppie StatoSilos-Lotto
1	L	E	StatoSilos	Lettura dei record StatoSilos -Lotto
5	S	R	Descritto	Inserimento coppie Lotto- FaseLotto
5	S	E	FaseLotto	Inserimento delle fasi del lotto
5	S	R	Avere	Inserimento della coppia FaseLotto - Parametro
1	S	R	Produzione	Inserimento della coppia Lotto - Laboratorio
5	S	R	Impiegato	Inserimento coppia Lotto -Dipendenti
40 (4000/100)	S	R	Lavorato	Inserimento coppia UnitaProdotto-Lotto
40 (4000/100)	S	E	UnitaProdotto	Inserimento record UnitaProdotto

Questa operazione viene effettuata 3 volte a settimana, di conseguenza si ottiene:

- Scrittura: $3 \times [1 + 5 + 5 + 5 + 1 + 5 + 40 + 40] = 306$ a settimana
- Lettura: $3 \times (1 + 1) = 6$ a settimana

8.2.6 OPERAZIONE 6 - Visualizzare servizi aggiuntivi per ogni prenotazione

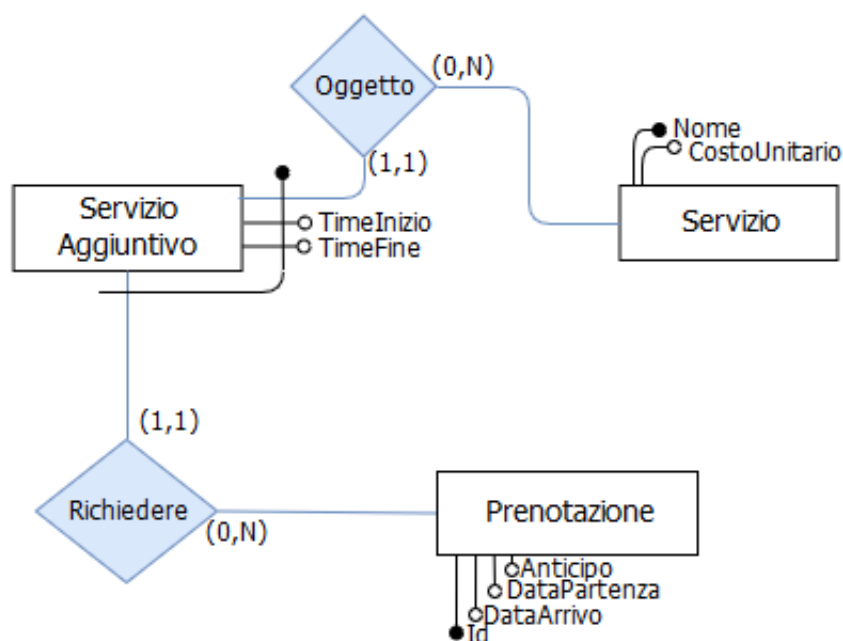


Tavola dei volumi		
Concetto	Tipo	Volume
Prenotazione	E	500
ServizioAggiuntivo	E	150
Servizio	E	12
Richiedere	R	150
Oggetto	R	150

Accessi	Tipo operazione	Tipo costruito	Nome costruito	Descrizione
500	L	E	Prenotazione	Lettura record Prenotazione
150	L	R	Richiedere	Lettura coppie Prenotazione - ServizioAggiuntivo
150	L	E	ServizioAggiuntivo	Lettura record ServizioAggiuntivo
150	L	R	Oggetto	Lettura coppie ServizioAggiuntivo - Servizio
12	L	E	Servizio	Lettura record servizio

Questa operazione viene effettuata 5 volte a settimana, di conseguenza si ottiene:

- Scrittura: 0
- Lettura: $5 \times [500 + 150 + 150 + 150 + 12] = 4810$ a settimana

8.2.7 OPERAZIONE 7 - Leggere i dati di ogni stato silos (incluso il livello di riempimento)

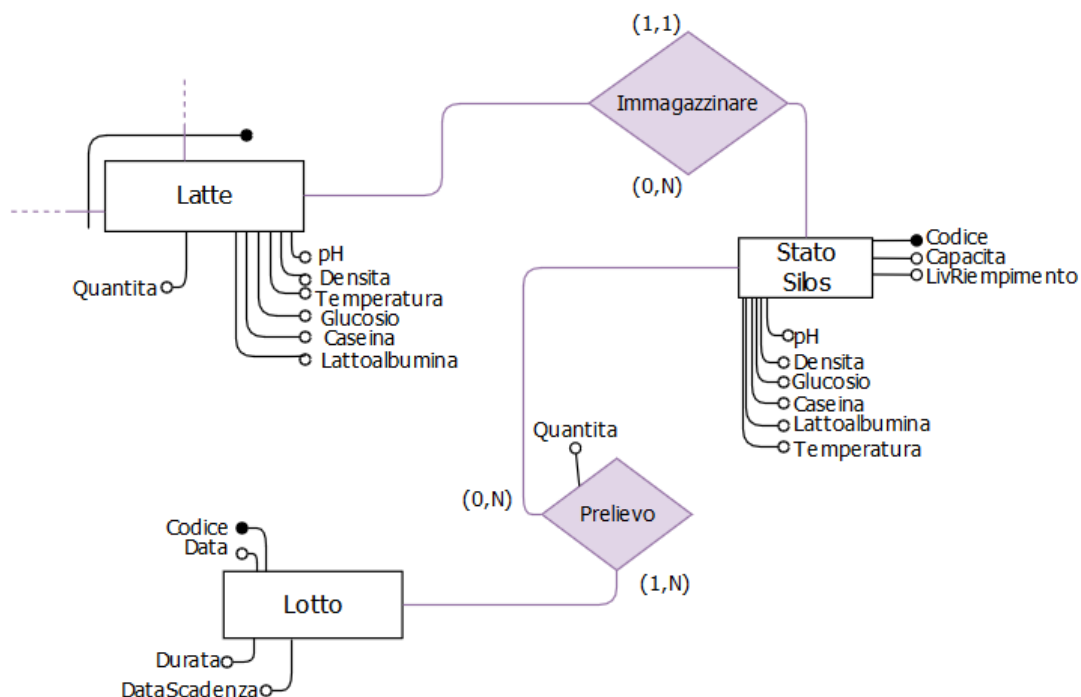


Tavola dei volumi		
Concetto	Tipo	Volume
StatoSilos	E	35
Latte	E	5000
Immagazzinare	R	5000
Prelievo	R	10000

Operazione 7 (Con ridondanza)				
Accessi	Tipo operazione	Tipo costruito	Nome costruito	Descrizione
35	L	E	StatoSilos	Lettura dei record del locale

Questa operazione viene effettuata 2 volte al giorno, di conseguenza si ottiene:

- Scrittura: 0
- Lettura: $2 \times 35 = 70$ al giorno

Operazione 7 (In assenza di ridondanza)				
Accessi	Tipo operazione	Tipo costruito	Nome costruito	Descrizione
35	L	E	StatoSilos	Lettura dei record dello StatoSilos
5000	L	R	Immagazzinare	Lettura delle coppie Latte - StatoSilos
5000	L	E	Latte	Lettura della quantità del latte inserito
10000	L	R	Prelievo	Lettura delle coppie StatoSilos – Lotto per leggere la quantità di latte prelevata

Questa operazione viene effettuata 2 volte al giorno, di conseguenza si ottiene:

- Scrittura: 0
- Lettura: $2 \times [35 + 5000 + 5000 + 10000] = 40070$ al giorno

In questo caso conviene mantenere la ridondanza, '*LivRiempimento*', all'interno della tabella **StatoSilos**.

8.2.8 OPERAZIONE 8 - Stampare l'importo di ogni pagamento

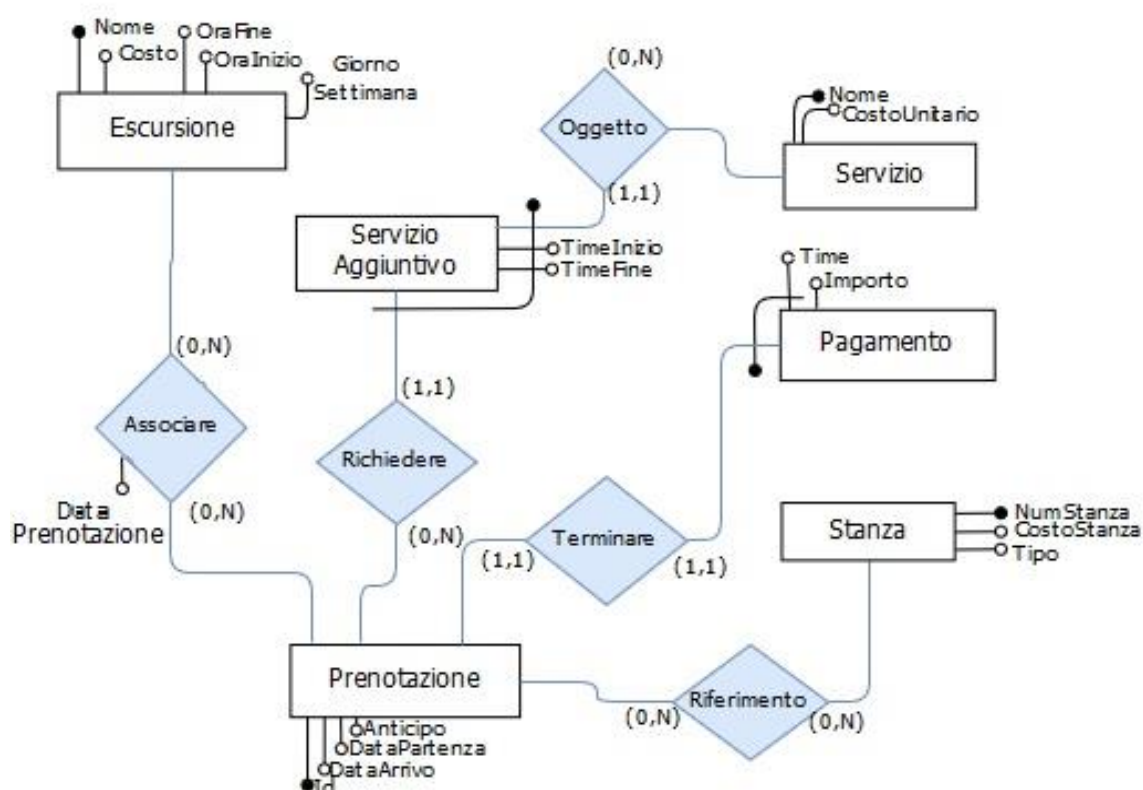


Tavola dei volumi		
Concetto	Tipo	Volume
Prenotazione	E	500
ServizioAggiuntivo	E	150
Servizio	E	12
Pagamento	E	500
Stanza	E	35
Escursione	E	50
Terminare	R	500
Riferimento	R	300
Richiedere	R	150
Oggetto	R	150
Associare	R	200

Operazione 8 (Con ridondanza)				
Accessi	Tipo operazione	Tipo costruito	Nome costruito	Descrizione
500	L	E	Pagamento	Lettura dei record di Pagamento

Questa operazione viene effettuata 5 volte al giorno, di conseguenza si ottiene:

- Scrittura: 0
- Lettura: $5 \times 500 = 2500$ al giorno

Operazione 8 (In assenza ridondanza)				
Accessi	Tipo operazione	Tipo costruito	Nome costruito	Descrizione
500	L	R	Terminare	Lettura delle coppie Pagamento-Prenotazione
500	L	E	Prenotazione	Lettura del record di Prenotazione
300	L	R	Riferimento	Lettura delle coppie Stanza - Prenotazione
35	L	E	Stanza	Lettura dei record di Stanza per ricavarne il prezzo giornaliero
150	L	R	Richiedere	Lettura delle coppie Prenotazione - ServizioAggiuntivo
150	L	E	ServizioAggiuntivo	Lettura record ServizioAggiuntivo
150	L	R	Oggetto	Lettur delle coppie ServizioAggiuntivo - Servizio
12	L	E	Servizio	Lettura dei record di Servizio per ricavarne il prezzo unitario
200	L	R	Associare	Lettura delle coppie Prenotazione – Escursione
50	L	E	Escursione	Lettura dei record Escursione per ricavarne il prezzo

Questa operazione viene effettuata 5 volte al giorno, di conseguenza si ottiene:

- Scrittura: 0
- Lettura: $5 \times [500 + 500 + 300 + 35 + 150 + 150 + 150 + 12 + 200 + 50] = 9485$ al giorno

In questo caso conviene mantenere la ridondanza, '*Importo*', all'interno della tabella **Pagamento**.

9. Traduzione modello logico

Area allevamento

Animale(Codice, Famiglia, DataNascita, Sesso, Specie, Razza, Altezza, Peso, DataArrivo, DataAcquisto, CodiceLocale, Fornitore);
Fornitore(PartitaIVA, Nome, Indirizzo, RagioneSociale);
GPS(ID, Longitudine, Latitudine, Time, Animale, Mungitrice);
Locale(Codice, PtCardinaleFinestra, TipoPavimento, Altezza, Larghezza, Lunghezza, CodiceStalla, Specie, NumMassimo, Allestimento);
Sensore(Codice, Temperatura, Umidita, LivAzoto, LivMetano, LivSporco, Locale);
Stalla(Codice, CodiceAgriturismo);
Agriturismo(Codice, Nome, Citta, Indirizzo);
Dipendente(Codice, Nome, Cognome, DataNascita, Tipologia, Gruppo, CodiceAgriturismo);
Gruppo(Codice);
Specie(Nome);
Pascolo(Codice, TimeInizio, TimeFine, NomeZona, CodiceAgriturismo, Locale);
Foraggio(Codice, Peso, KCal, Frutta, Cereali, Piante, Proteine, Glucidi, Fibre, Tipo);
Mangiatoia(Codice, Allestimento);
Abbeveratoio(Codice, Allestimento);
Condizionamentolluminazione(Codice, Tipo, Allestimento);
Allestimento(Codice);
Riproduzione(Codice, Time, Stato, Veterinario);
Veterinario(Matricola, Nome, Cognome, Parcella);
InterventoDiControllo(Codice, DataProgrammata, DataControllo, Stato, Esito, Veterinario, CodiceRiproduzione, Terapia);
Esame(Codice, Time, DescrizioneProcedimento, Nome, Macchinario, InterventoDiControllo);
Macchinario(Codice);
Recinzione(ID, X1, Y1, X2, Y2, NomeZona, CodiceAgriturismo);
SchedaDiGestazione(CodiceRiproduzione, Veterinario);
Zona(CodiceAgriturismo, Nome);
StatoSensore(Time, Sensore, Stato);
StatoMangiatoia(Mangiatoia, Istante, Quantita, Foraggio);
StatoAbbeveratoio(Abbeveratoio, Istante, SaliMinerali, Vitamine, Quantita);
Andare(Animale, Pascolo);
Effettua(Animale, Riproduzione);
Parentela(Genitore, Figlio);

Vincoli di Integrità referenziale:

- 1) Esiste il vincolo di integrità referenziale tra l'attributo CodiceLocale della relazione Animale e l'attributo Codice della relazione Locale
- 2) Esiste il vincolo di integrità referenziale tra l'attributo CodiceStalla della relazione Locale e l'attributo Codice della relazione Stalla.
- 3) Esiste il vincolo di integrità referenziale tra l'attributo CodiceAgriturismo della relazione Stalla e l'attributo Codice della relazione Agriturismo.
- 4) Esiste il vincolo di integrità referenziale tra l'attributo CodiceAgriturismo della relazione Zona e l'attributo Codice della relazione Agriturismo.

- 5) Esiste il vincolo di integrità referenziale tra l'attributo NomeZona della relazione Pascolo e l'attributo Nome della relazione Zona.
- 6) Esiste il vincolo di integrità referenziale tra l'attributo Matricola della relazione Veterinario e l'attributo Veterinario della relazione Riproduzione.
- 7) Esiste il vincolo di integrità referenziale tra l'attributo Matricola della relazione Veterinario e l'attributo Veterinario della relazione InterventoDiControllo.
- 8) Esiste il vincolo di integrità referenziale tra l'attributo Codice della relazione Riproduzione e l'attributo CodiceRiproduzione della relazione InterventoDiControllo.
- 9) Esiste il vincolo di integrità referenziale tra l'attributo PartitaIVA della relazione Fornitore e l'attributo Fornitore della relazione Animale.
- 10) Esiste il vincolo di integrità referenziale tra l'attributo Nome della relazione Specie e l'attributo Specie della relazione Locale.
- 11) Esiste il vincolo di integrità referenziale tra l'attributo Codice della relazione Locale e l'attributo Locale della relazione Sensore.
- 12) Esiste il vincolo di integrità referenziale tra l'attributo Codice della relazione Allestimento e l'attributo Allestimento della relazione Mangiatoia.
- 13) Esiste il vincolo di integrità referenziale tra l'attributo Codice della relazione Allestimento e l'attributo Allestimento della relazione Abbeveratoio.
- 14) Esiste il vincolo di integrità referenziale tra l'attributo Codice della relazione Allestimento e l'attributo Allestimento della relazione CondizionamentoIlluminazione.
- 15) Esiste il vincolo di integrità referenziale tra l'attributo Codice della relazione Foraggio e l'attributo Foraggio della relazione StatoMangiatoia.
- 16) Esiste il vincolo di integrità referenziale tra l'attributo Mangiatoia della relazione Mangiatoia e l'attributo Mangiatoia della relazione StatoMangiatoia.
- 17) Esiste il vincolo di integrità referenziale tra l'attributo Codice della relazione Abbeveratoio e l'attributo Abbeveratoio della relazione StatoAbbeveratoio.
- 18) Esiste il vincolo di integrità referenziale tra l'attributo Codice della relazione Gruppo e l'attributo Gruppo della relazione Dipendente.
- 19) Esiste il vincolo di integrità referenziale tra l'attributo CodiceAgriturismo della relazione Dipendente e l'attributo Codice della relazione Agriturismo.
- 20) Esiste il vincolo di integrità referenziale tra l'attributo Nome della relazione Zona e l'attributo NomeZona della relazione Recinzione.
- 21) Esiste il vincolo di integrità referenziale tra l'attributo Animale della relazione Andare e l'attributo Codice della relazione Animale.
- 22) Esiste il vincolo di integrità referenziale tra l'attributo Pascolo della relazione Andare e l'attributo Codice della relazione Pascolo.
- 23) Esiste il vincolo di integrità referenziale tra l'attributo Animale della relazione Effettua e l'attributo Codice della relazione Animale.
- 24) Esiste il vincolo di integrità referenziale tra l'attributo Riproduzione della relazione Effettua e l'attributo Codice della relazione Riproduzione.
- 25) Esiste il vincolo di integrità referenziale tra l'attributo GPS della relazione Animale e l'attributo ID della relazione GPS.
- 26) Esiste il vincolo di integrità referenziale tra l'attributo Allestimento della relazione Locale e l'attributo Codice della relazione Allestimento.
- 27) Esiste il vincolo di integrità referenziale tra l'attributo Macchinario della relazione Esame e l'attributo Codice della relazione Macchinario.

- 28) Esiste il vincolo di integrità referenziale tra l'attributo InterventoDiControllo della relazione Esame e l'attributo Codice della relazione InterventoDiControllo.
- 29) Esiste il vincolo di integrità referenziale tra l'attributo Terapia della relazione InterventoDiControllo e l'attributo Codice della relazione Terapia.
- 30) Esiste il vincolo di integrità referenziale tra l'attributo Sensore della relazione StatoSensore e l'attributo Codice della relazione Sensore.
- 31) Esiste il vincolo di integrità referenziale tra l'attributo Matricola della relazione Veterinario e l'attributo Veterinario della relazione SchedaDiGestazione.
- 32) Esiste il vincolo di integrità referenziale tra l'attributo Specie della relazione Animale e l'attributo Nome della relazione Specie.
- 33) Esiste il vincolo di integrità referenziale tra l'attributo CodiceAgriturismo della relazione Pascolo e l'attributo CodiceAgriturismo della relazione Zona.
- 34) Esiste il vincolo di integrità referenziale tra l'attributo CodiceAgriturismo della relazione Recinzione e l'attributo CodiceAgriturismo della relazione Zona.
- 35) Esiste il vincolo di integrità referenziale tra l'attributo Figlio della relazione Parentela e l'attributo Codice della relazione Animale.
- 36) Esiste il vincolo di integrità referenziale tra l'attributo Genitore della relazione Parentela e l'attributo Codice della relazione Animale
- 37) Esiste il vincolo di integrità referenziale tra l'attributo Locale della relazione Pascolo e l'attributo Codice della relazione Locale
- 38) Esiste il vincolo di integrità referenziale tra l'attributo Animale della relazione GPS e l'attributo Codice della relazione Animale
- 39) Esiste il vincolo di integrità referenziale tra l'attributo Mungitrice della relazione GPS e l'attributo Codice della relazione Mungitrice

Area healthcare

Terapia(Codice, Veterinario, Animale, Data, DataInizioTerapia, Durata, Posologia, Patologia);

Patologia(NomePatologia);

IndicatoriSoggettivi(Codice, Entita, Tipo, PtCorpo, NomeDisturbo);

Orario(Ora);

Quarantena(Time, Animale);

SchedaMedica(Time, Animale);

Farmaco(NomeCommerciale, Patologia, PrincipioAttivo);

SchedaTerapia(Farmaco, Patologia, DataInizio, Terapia, DataFine);

Indice(Codice, Nome);

Visita(Veterinario, Animale, Data);

Dosaggio(Farmaco, Patologia, DataInizio, Terapia, Ora, Dose);

Possibilita(IndicatoreSoggettivo, Time, Animale);

Valutare(Indice, Time, Animale, Valore);

Vincoli di Integrità referenziale:

- 1) Esiste il vincolo di integrità referenziale tra l'attributo Animale della relazione Quarantena e l'attributo Codice della relazione Animale.
- 2) Esiste il vincolo di integrità referenziale tra l'attributo Animale della relazione SchedaMedica e l'attributo Codice della relazione Animale.
- 3) Esiste il vincolo di integrità referenziale tra l'attributo Patologia della relazione Farmaco e l'attributo NomePatologia della relazione Patologia.
- 4) Esiste il vincolo di integrità referenziale tra l'attributo Farmaco della relazione SchedaTerapia e l'attributo NomeCommerciale della relazione Farmaco.
- 5) Esiste il vincolo di integrità referenziale tra l'attributo Patologia della relazione SchedaTerapia e l'attributo Patologia della relazione Farmaco.
- 6) Esiste il vincolo di integrità referenziale tra l'attributo Terapia della relazione SchedaTerapia e l'attributo Codice della relazione Terapia.
- 7) Esiste il vincolo di integrità referenziale tra l'attributo Veterinario della relazione Visita e l'attributo Matricola della relazione Veterinario.
- 8) Esiste il vincolo di integrità referenziale tra l'attributo Veterinario della relazione Terapia e l'attributo Veterinario della relazione Visita.
- 9) Esiste il vincolo di integrità referenziale tra l'attributo Animale della relazione Visita e l'attributo Codice della relazione Animale.
- 10) Esiste il vincolo di integrità referenziale tra l'attributo Patologia della relazione Terapia e l'attributo NomePatologia della relazione Patologia.
- 11) Esiste il vincolo di integrità referenziale tra l'attributo Veterinario della relazione Terapia e l'attributo Matricola della relazione Veterinario.
- 12) Esiste il vincolo di integrità referenziale tra l'attributo Patologia della relazione Dosaggio e l'attributo Patologia della relazione SchedaTerapia.
- 13) Esiste il vincolo di integrità referenziale tra l'attributo Farmaco della relazione Dosaggio e l'attributo Farmaco della relazione SchedaTerapia.

- 14) Esiste il vincolo di integrità referenziale tra l'attributo Terapia della relazione Dosaggio e l'attributo Terapia della relazione SchedaTerapia.
- 15) Esiste il vincolo di integrità referenziale tra l'attributo Ora della relazione Dosaggio e l'attributo Ora della relazione Orario.
- 16) Esiste il vincolo di integrità referenziale tra l'attributo IndicatoreSoggettivo della relazione Possibilita e l'attributo Codice della relazione IndicatoriSoggettivi.
- 17) Esiste il vincolo di integrità referenziale tra l'attributo Time della relazione Possibilita e l'attributo Time della relazione SchedaMedica.
- 18) Esiste il vincolo di integrità referenziale tra l'attributo Animale della relazione Possibilita e l'attributo Animale della relazione SchedaMedica.
- 19) Esiste il vincolo di integrità referenziale tra l'attributo Animale della relazione Terapia e l'attributo Animale della relazione Visita.
- 20) Esiste il vincolo di integrità referenziale tra l'attributo Data della relazione Terapia e l'attributo Data della relazione Visita.
- 21) Esiste il vincolo di integrità referenziale tra l'attributo Indice della relazione Valutare e l'attributo Codice della relazione Indice.
- 22) Esiste il vincolo di integrità referenziale tra l'attributo Time della relazione Valutare e l'attributo Time della relazione SchedaMedica.
- 23) Esiste il vincolo di integrità referenziale tra l'attributo Animale della relazione Valutare e l'attributo Animale della relazione SchedaMedica.

Area Produzione

Silos(Codice, Marca);

StatoSilos(Codice, Silos, Capacita, LivRiempimento, pH, Densita, Glucosio, Caseina, Lattoalbumina, Temperatura);

ParametroIdeale(Nome);

ProdottoCaseario(Nome, ZonaOrigine, Tipo, GradoDeperibilita, Costo);

UnitaProdotto(Codice, Peso, ScaffaleCantina, ScaffaleMagazzino, DataInizio, ProdottoCaseario, Lotto, Ordine);

ScaffaleCantina(Codice, CapienzaMax, Cantina);

ScaffaleMagazzino(Codice, CapienzaMax, Magazzino);

Cantina(Codice, Agriturismo);

Magazzino(Codice, Agriturismo);

StatoCantina(Time, Cantina, Temperatura, Umidita);

Lotto(Codice, Data, Durata, DataScadenza, Laboratorio, Dipendente);

FaseLotto(Lotto, NumFase, Durata, Temperatura);

Parametro(Nome);

Mungitrice(Codice, Modello, Marca);

Mungitura(Mungitrice, Time);

Latte(Mungitrice, Time, Animale, pH, Densita, Glucosio, Caseina, Lattoalbumina, Temperatura, StatoSilos);

Ricetta(Prodotto, Descrizione, DataStagionatura);

Fase(Prodotto, IDFase, Durata);

Laboratorio(Codice, Tipo, Agriturismo);

Prelievo(Lotto, StatoSilos, Quantita);

Avere(Lotto, NumFase, Parametro, Quantita);

Composta(Prodotto, IdFase, Parametro, Quantita);

Impiegato(Lotto, Dipendente);

Vincoli di Integrità referenziale:

- 1) Esiste il vincolo di integrità referenziale tra l'attributo Mungitrice della relazione Latte e l'attributo Mungitrice della relazione Mungitura.
- 2) Esiste il vincolo di integrità referenziale tra l'attributo Time della relazione Latte e l'attributo Time della relazione Mungitura.
- 3) Esiste il vincolo di integrità referenziale tra l'attributo Animale della relazione Latte e l'attributo Codice della relazione Animale.
- 4) Esiste il vincolo di integrità referenziale tra l'attributo Mungitrice della relazione Mungitura e l'attributo Codice della relazione Mungitrice.
- 5) Esiste il vincolo di integrità referenziale tra l'attributo StatoSilos della relazione Latte e l'attributo Codice della relazione StatoSilos.
- 6) Esiste il vincolo di integrità referenziale tra l'attributo GPS della relazione Mungitrice e l'attributo ID della relazione GPS.
- 7) Esiste il vincolo di integrità referenziale tra l'attributo Silos della relazione StatoSilos e l'attributo Codice della relazione Silos.
- 8) Esiste il vincolo di integrità referenziale tra l'attributo StatoSilos della relazione Prelievo e l'attributo Codice della relazione StatoSilos.
- 9) Esiste il vincolo di integrità referenziale tra l'attributo Lotto della relazione Prelievo e l'attributo Codice della relazione Lotto.

- 10) Esiste il vincolo di integrità referenziale tra l'attributo Agriturismo della relazione Cantina e l'attributo Codice della relazione Agriturismo.
- 11) Esiste il vincolo di integrità referenziale tra l'attributo Cantina della relazione StatoCantina e l'attributo Codice della relazione Cantina.
- 12) Esiste il vincolo di integrità referenziale tra l'attributo Agriturismo della relazione Magazzino e l'attributo Codice della relazione Agriturismo
- 13) Esiste il vincolo di integrità referenziale tra l'attributo Cantina della relazione ScaffaleCantina e l'attributo Codice della relazione Cantina
- 14) Esiste il vincolo di integrità referenziale tra l'attributo Magazzino della relazione ScaffaleMagazzino e l'attributo Codice della relazione Magazzino
- 15) Esiste il vincolo di integrità referenziale tra l'attributo ScaffaleCantina della relazione UnitaProdotto e l'attributo Codice della relazione ScaffaleCantina
- 16) Esiste il vincolo di integrità referenziale tra l'attributo ScaffaleMagazzino della relazione UnitaProdotto e l'attributo Codice della relazione ScaffaleMagazzino
- 17) Esiste il vincolo di integrità referenziale tra l'attributo ProdottoCaseario della relazione UnitaProdotto e l'attributo Nome della relazione ProdottoCaseario.
- 18) Esiste il vincolo di integrità referenziale tra l'attributo Agriturismo della relazione Laboratorio e l'attributo Codice della relazione Agriturismo
- 19) Esiste il vincolo di integrità referenziale tra l'attributo Lotto della relazione UnitaProdotto e l'attributo Codice della relazione Lotto
- 20) Esiste il vincolo di integrità referenziale tra l'attributo Laboratorio della relazione Lotto e l'attributo Codice della relazione Laboratorio
- 21) Esiste il vincolo di integrità referenziale tra l'attributo Dipendente della relazione Lotto e l'attributo Codice della relazione Dipendente
- 22) Esiste il vincolo di integrità referenziale tra l'attributo Lotto della relazione FaseLotto e l'attributo Codice della relazione Lotto.
- 23) Esiste il vincolo di integrità referenziale tra l'attributo Parametro della relazione Avere e l'attributo Nome della relazione Parametro
- 24) Esiste il vincolo di integrità referenziale tra l'attributo NumFase della relazione Avere e l'attributo NumFase della relazione FaseLotto
- 25) Esiste il vincolo di integrità referenziale tra l'attributo Lotto della relazione Avere e l'attributo Lotto della relazione FaseLotto
- 26) Esiste il vincolo di integrità referenziale tra l'attributo Prodotto della relazione Ricetta e l'attributo Nome della relazione ProdottoCaseario
- 27) Esiste il vincolo di integrità referenziale tra l'attributo Prodotto della relazione Fase e l'attributo Prodotto della relazione Ricetta
- 28) Esiste il vincolo di integrità referenziale tra l'attributo Parametro della relazione Composta e l'attributo Nome della relazione ParametroIdeale
- 29) Esiste il vincolo di integrità referenziale tra l'attributo Prodotto della relazione Composta e l'attributo Prodotto della relazione Fase
- 30) Esiste il vincolo di integrità referenziale tra l'attributo IdFase della relazione Composta e l'attributo IdFase della relazione Fase.
- 31) Esiste il vincolo di integrità referenziale tra l'attributo Lotto della relazione Impiegato e l'attributo Codice della relazione Lotto
- 32) Esiste il vincolo di integrità referenziale tra l'attributo Dipendente della relazione Impiegato e l'attributo Codice della relazione Dipendente

Area Soggiorno

Prenotazione(ID, DataArrivo, DataPartenza, Anticipo, UtenteRegistrato, UtenteNonRegistrato);

Escursione(Nome, Costo, Itinerario, OraFine, OraInizio, GiornoSettimana, Guida);

Servizio(Nome, CostoUnitario);

Itinerario(Codice);

Area(Itinerario, Ora, Nome, DurataSosta, Agriturismo);

ServizioAggiuntivo(Servizio, TimeInizio, TimeFine, Prenotazione);

Stanza(NumStanza, CostoStanza, Tipo);

Letto(CodLetto, TipoLetto, Stanza);

Pagamento(Prenotazione, Time, Importo);

Registrato(Codice, Telefono, Nome, Cognome, CodCarta, Tipologia, NumDocumento, Scadenza, Ente);

NonRegistrato(Codice, CodiceCarta);

Riferimento(Prenotazione, Stanza);

Associare(Prenotazione, Escursione, DataPrenotazione);

Vincoli di Integrità referenziale:

- 1) Esiste il vincolo di integrità referenziale tra l'attributo Itinerario della relazione Area e l'attributo Codice della relazione Itinerario
- 2) Esiste il vincolo di integrità referenziale tra l'attributo Servizio della relazione ServizioAggiuntivo e l'attributo Nome della relazione Servizio
- 3) Esiste il vincolo di integrità referenziale tra l'attributo Prenotazione della relazione ServizioAggiuntivo e l'attributo ID della relazione Prenotazione.
- 4) Esiste il vincolo di integrità referenziale tra l'attributo Prenotazione della relazione Pagamento e l'attributo ID della relazione Prenotazione
- 5) Esiste il vincolo di integrità referenziale tra l'attributo Stanza della relazione Letto e l'attributo NumStanza della relazione Stanza
- 6) Esiste il vincolo di integrità referenziale tra l'attributo Itinerario della relazione Escursione e l'attributo Codice della relazione Itinerario
- 7) Esiste il vincolo di integrità referenziale tra l'attributo Agriturismo della relazione Area e l'attributo Codice della relazione Agriturismo
- 8) Esiste il vincolo di integrità referenziale tra l'attributo UtenteRegistrato della relazione Prenotazione e l'attributo Codice della relazione Registrato
- 9) Esiste il vincolo di integrità referenziale tra l'attributo UtenteNonRegistrato della relazione Prenotazione e l'attributo Codice della relazione NonRegistrato
- 10) Esiste il vincolo di integrità referenziale tra l'attributo Prenotazione della relazione Riferimento e l'attributo ID della relazione Prenotazione
- 11) Esiste il vincolo di integrità referenziale tra l'attributo Stanza della relazione Riferimento e l'attributo NumStanza della relazione Stanza
- 12) Esiste il vincolo di integrità referenziale tra l'attributo Prenotazione della relazione Associare e l'attributo ID della relazione Prenotazione
- 13) Esiste il vincolo di integrità referenziale tra l'attributo Escursione della relazione Associare e l'attributo Nome della relazione Escursione
- 14) Esiste il vincolo di integrità referenziale tra l'attributo Guida della relazione Escursione e l'attributo Codice della relazione Dipendente

Area Store

Account(NomeUtente, Password, DataIscrizione, Domanda, Risposta);

Ordine(Codice, Stato, Time, Account);

Recensione(Codice, Testo, Conservazione, Gradimento, Qualita, Gusto, UnitaProdotto, Data);

Reso (Codice, DataReso, UnitaProdotto);

Spedizione (Codice, Stato);

ParametriQualita (Nome);

Hub(Codice, Citta, Indirizzo);

Trasporto(UnitaProdotto, Spedizione, DataPrevista, DataEffettiva);

Controllo(Reso, ParametroQualita, Valutazione);

Prevedere(Spedizione, Hub, Time);

Selezione(ProdottoCaseario, Ordine, Quantita);

Vincoli di Integrità referenziale:

- 1) Esiste il vincolo di integrità referenziale tra l'attributo UnitaProdotto della relazione Trasporto e l'attributo Codice della relazione UnitaProdotto
- 2) Esiste il vincolo di integrità referenziale tra l'attributo Spedizione della relazione Trasporto e l'attributo Codice della relazione Spedizione
- 3) Esiste il vincolo di integrità referenziale tra l'attributo Reso della relazione Controllo e l'attributo Codice della relazione Reso
- 4) Esiste il vincolo di integrità referenziale tra l'attributo ParametroQualita della relazione Controllo e l'attributo Nome della relazione ParametroQualita
- 5) Esiste il vincolo di integrità referenziale tra l'attributo Spedizione della relazione Prevedere e l'attributo Codice della relazione Spedizione
- 6) Esiste il vincolo di integrità referenziale tra l'attributo Hub della relazione Prevedere e l'attributo Codice della relazione Hub
- 7) Esiste il vincolo di integrità referenziale tra l'attributo Ordine della relazione Selezione e l'attributo Codice della relazione Ordine
- 8) Esiste il vincolo di integrità referenziale tra l'attributo ProdottoCaseario della relazione Selezione e l'attributo Nome della relazione ProdottoCaseario
- 9) Esiste il vincolo di integrità referenziale tra l'attributo Ordine della relazione UnitaProdotto e l'attributo Codice della relazione Ordine
- 10) Esiste il vincolo di integrità referenziale tra l'attributo Account della relazione Ordine e l'attributo NomeUtente della relazione Account
- 11) Esiste il vincolo di integrità referenziale tra l'attributo UnitaProdotto della relazione Reso e l'attributo Codice della relazione UnitaProdotto
- 12) Esiste il vincolo di integrità referenziale tra l'attributo UnitaProdotto della relazione Recensione e l'attributo Codice della relazione UnitaProdotto

10. Dipendenze Funzionali

10.1 Area Allevamento

Animale(Codice, Famiglia, DataNascita, Sesso, Razza, Altezza, Peso, DataArrivo, DataAcquisto, CodiceLocale, Fornitore);

Codice → Famiglia, DataNascita, Sesso, Razza, Altezza, Peso, DataArrivo, DataAcquisto, CodiceLocale, Fornitore.

Poiché l'attributo a sinistra è superchiave, Animale è in BCNF, quindi non necessita di ulteriori cambiamenti.

Fornitore(PartitaIVA, Nome, Indirizzo, RagioneSociale);

PartitaIVA → Nome, Indirizzo, RagioneSociale

Poiché l'attributo a sinistra è superchiave, Fornitore è in BCNF, quindi non necessita di ulteriori cambiamenti.

GPS(ID, Longitudine, Latitudine, Time, Animale, Mungitrice);

ID → Longitudine, Latitudine, Time, Animale, Mungitrice

Poiché l'attributo a sinistra è superchiave, GPS è in BCNF, quindi non necessita di ulteriori cambiamenti.

Locale(Codice, PtCardinaleFinestra, TipoPavimento, Altezza, Larghezza, Lunghezza, CodiceStalla, Specie, NumMassimo, Allestimento);

Codice → PtCardinaleFinestra, TipoPavimento, Altezza, Larghezza, Lunghezza, CodiceStalla, Specie, NumMassimo, Allestimento

Poiché l'attributo a sinistra è superchiave, Locale è in BCNF, quindi non necessita di ulteriori cambiamenti.

Sensore(Codice, Temperatura, Umidità, LivAzoto, LivMetano, LivSporco, Locale);

Codice → Temperatura, Umidità, LivAzoto, LivMetano, LivSporco, Locale

Poiché l'attributo a sinistra è superchiave, Sensore è in BCNF, quindi non necessita di ulteriori cambiamenti.

Stalla(Codice, CodiceAgriturismo);

Codice → CodiceAgriturismo

Poiché l'attributo a sinistra è superchiave, Stalla è in BCNF, quindi non necessita di ulteriori cambiamenti.

Agriturismo(Codice, Nome, Citta, Indirizzo);

Codice → Nome, Citta, Indirizzo

Dipendente(Codice, Nome, Cognome, DataNascita, Tipologia, Gruppo, CodiceAgriturismo);

Codice → Nome, Cognome, DataNascita, Tipologia, Gruppo, CodiceAgriturismo

Poiché l'attributo a sinistra è superchiave, Dipendente è in BCNF, quindi non necessita di ulteriori cambiamenti.

Gruppo(Codice);

Poiché non esistono dipendenze funzionali, Gruppo è in BCNF.

Specie(Nome);

Poiché non esistono dipendenze funzionali, Specie è in BCNF.

Pascolo(Codice, TimeInizio, TimeFine, NomeZona, CodiceAgriturismo, Locale);

Codice → TimeInizio, TimeFine, NomeZona, CodiceAgriturismo, Locale

Foraggio(Codice, Peso, KCal, Frutta, Cereali, Piante, Proteine, Glucidi, Fibre, Tipo);

Codice → Peso, KCal, Frutta, Cereali, Piante, Proteine, Glucidi, Fibre, Tipo
Peso, Frutta, Cereali, Piante, Proteine, Glucidi, Fibre → KCal

Poiché l'attributo a sinistra è superchiave, Foraggio è in BCNF, quindi non necessita di ulteriori cambiamenti.

Mangiatoia(Codice, Allestimento);

Codice → Allestimento

Poiché l'attributo a sinistra è superchiave, Mangiatoia è in BCNF, quindi non necessita di ulteriori cambiamenti.

Abbeveratoio(Codice, Allestimento);

Codice → Allestimento

Poiché l'attributo a sinistra è superchiave, Abbeveratoio è in BCNF, quindi non necessita di ulteriori cambiamenti.

Condizionamentolluminazione(Codice, Tipo, Allestimento);

Codice → Tipo, Allestimento

Poiché l'attributo a sinistra è superchiave, Condizionamentolluminazione è in BCNF, quindi non necessita di ulteriori cambiamenti.

Allestimento(Codice);

Poiché non esistono dipendenze funzionali, Allestimento è in BCNF.

Riproduzione(Codice, Time, Stato, Veterinario);

Codice → Time, Stato, Veterinario

Veterinario(Matricola, Nome, Cognome, Parcella);

Matricola → Nome, Cognome, Parcella

Poiché l'attributo a sinistra è superchiave, Veterinario è in BCNF, quindi non necessita di ulteriori cambiamenti.

InterventoDiControllo(Codice, DataProgrammata, DataControllo, Stato, Esito, Veterinario, CodiceRiproduzione, Terapia);

Codice → DataProgrammata, DataControllo, Stato, Esito, Veterinario, CodiceRiproduzione, Terapia

Poiché l'attributo a sinistra è superchiave, InterventoDiControllo è in BCNF, quindi non necessita di ulteriori cambiamenti.

Esame(Codice, Time, DescrizioneProcedimento, Nome, Macchinario, InterventoDiControllo);

Codice → Time, DescrizioneProcedimento, Nome, Macchinario, InterventoDiControllo

Poiché l'attributo a sinistra è superchiave, Esame è in BCNF, quindi non

necessita di ulteriori cambiamenti.

Macchinario(Codice);

Poiché non esistono dipendenze funzionali, Macchinario è in BCNF.

Recinzione(ID, X1, Y1, X2, Y2, NomeZona, CodiceAgriturismo);

$ID \rightarrow X1, Y1, X2, Y2, \text{NomeZona}, \text{CodiceAgriturismo}$

Poiché l'attributo a sinistra è superchiave, Recinzione è in BCNF, quindi non necessita di ulteriori cambiamenti.

SchedaDiGestazione(CodiceRiproduzione);

Poiché non esistono dipendenze funzionali, SchedaDiGestazione è in BCNF.

Zona(CodiceAgriturismo, Nome);

Poiché non esistono dipendenze funzionali, Zona è in BCNF.

StatoSensore(Time, Sensore, Stato);

$\text{Time}, \text{Sensore} \rightarrow \text{Stato}$

Poiché gli attributi a sinistra sono superchiave, StatoSensore è in BCNF, quindi non necessita di ulteriori cambiamenti.

StatoMangiatoia(Mangiatoia, Istante, Quantita, Foraggio);

$\text{Mangiatoia}, \text{Istante} \rightarrow \text{Quantita}, \text{Foraggio}$

Poiché gli attributi a sinistra sono superchiave, StatoMangiatoia è in BCNF, quindi non necessita di ulteriori cambiamenti.

StatoAbbeveratoio(Abbeveratoio, Istante, SaliMinerali, Vitamine, Quantita);

$\text{Abbeveratoio}, \text{Istante} \rightarrow \text{SaliMinerali}, \text{Vitamine}, \text{Quantita}$

Poiché gli attributi a sinistra sono superchiave, StatoAbbeveratoio è in BCNF, quindi non necessita di ulteriori cambiamenti.

Andare(Animale, Pascolo);

Poiché non esistono dipendenze funzionali, Andare è in BCNF.

Effettua(Animale, Riproduzione);

Poiché non esistono dipendenze funzionali, Effettua è in BCNF.

Parentela(Genitore, Figlio);

Poiché non esistono dipendenze funzionali, Parentela è in BCNF.

- Area healthcare

Terapia(Codice, Veterinario, Animale, Data, DataInizioTerapia, Durata, Posologia, Patologia, Veterinario);

Codice → Veterinario, Animale, Data, DataInizioTerapia, Durata, Posologia, Patologia, Veterinario

Poiché l'attributo a sinistra è superchiave, Terapia è BCFN, quindi non necessita di ulteriori cambiamenti

Patologia(NomePatologia);

Poiché non esistono dipendenze funzionali, Patologia è in BCNF.

IndicatoriSoggettivi(Codice, Entita, Tipo, PtCorpo, NomeDisturbo);

Codice → Entita, Tipo, PtCorpo, NomeDisturbo

Poiché l'attributo a sinistra è superchiave, IndicatoriSoggettivi è BCFN, quindi non necessita di ulteriori cambiamenti

Orario(Ora);

Poiché non esistono dipendenze funzionali, Orario è in BCNF.

Quarantena(Time, Animale);

Poiché non esistono dipendenze funzionali, Quarantela è in BCNF.

SchedaMedica(Time, Animale);

Poiché non esistono dipendenze funzionali, SchedaMedica è in BCNF.

Farmaco(NomeCommerciale, Patologia, PrincipioAttivo);

NomeCommerciale, Patologia → PrincipioAttivo

Poiché l'attributo a sinistra è superchiave, Farmaco è BCFN, quindi non necessita di ulteriori cambiamenti

SchedaTerapia(Farmaco, Patologia, DataInizio, Terapia, DataFine);

Farmaco, Patologia, DataInizio, Terapia → DataFine

Poiché l'attributo a sinistra è superchiave, SchedaTerapia è in BCFN, quindi non necessita di ulteriori cambiamenti

Visita(Veterinario, Animale, Data);

Poiché non esistono dipendenze funzionali, Visita è in BCNF.

Dosaggio(Farmaco, Patologia, DataInizio, Terapia, Ora, Dose);

Farmaco, Patologia, DataInizio, Terapia, Ora → Dose

Possibilita(IndicatoreSoggettivo, Time, Animale);

Poiché non esistono dipendenze funzionali, Possibilita è in BCNF.

Valutare(Indice, Time, Animale, Valore);

Indice, Time, Animale → Valore

Poiché l'attributo a sinistra è superchiave, Valutare è in BCFN, quindi non necessita di ulteriori cambiamenti

Indice (Codice, Nome);

Codice → Nome

Poiché l'attributo a sinistra è superchiave, Indice è in BCFN, quindi non necessita di ulteriori cambiamenti

- Area Produzione

Silos(Codice, Marca);

Codice → Marca

Poiché l'attributo a sinistra è superchiave, Silos è in BCNF, quindi non necessita di ulteriori cambiamenti.

StatoSilos(Codice, Silos, Capacita, LivRiempimento, pH, Densita, Glucosio, Caseina, Lattoalbumina, Temperatura);

Codice → Silos, Capacita, LivRiempimento, pH, Densita, Glucosio, Caseina, Lattoalbumina, Temperatura

Poiché l'attributo a sinistra è superchiave, StatoSilos è in BCNF, quindi non necessita di ulteriori cambiamenti.

ParametroIdeale(Nome);

Poiché non esistono dipendenze funzionali, ParametroIdeale è in BCNF.

ProdottoCaseario(Nome, ZonaOrigine, Tipo, GradoDeperibilita, Costo);

Nome → ZonaOrigine, Tipo, GradoDeperibilita, Costo

Poiché l'attributo a sinistra è superchiave, ProdottoCaseario è in BCNF, quindi non necessita di ulteriori cambiamenti.

UnitaProdotto(Codice, Peso, ScaffaleCantina, ScaffaleMagazzino, DataInizio, ProdottoCaseario, Lotto, Ordine);

Codice → , Peso, ScaffaleCantina, ScaffaleMagazzino, DataInizio, ProdottoCaseario, Lotto, Ordine

Poiché l'attributo a sinistra è superchiave, UnitaProdotto è in BCNF, quindi non necessita di ulteriori cambiamenti.

ScaffaleCantina(Codice, CapienzaMax, Cantina);

Codice → CapienzaMax, Cantina

Poiché l'attributo a sinistra è superchiave, ScaffaleCantina è in BCNF, quindi non necessita di ulteriori cambiamenti.

ScaffaleMagazzino(Codice, CapienzaMax, Magazzino);

Codice → CapienzaMax, Magazzino

Poiché l'attributo a sinistra è superchiave, ScaffaleMagazzino è in BCNF, quindi non necessita di ulteriori cambiamenti.

Cantina(Codice, Agriturismo);

Codice → Agriturismo

Poiché l'attributo a sinistra è superchiave, Cantina è in BCNF, quindi non necessita di ulteriori cambiamenti.

Magazzino(Codice, Agriturismo);

Codice → Agriturismo

Poiché l'attributo a sinistra è superchiave, Magazzino è in BCNF, quindi non necessita di ulteriori cambiamenti.

StatoCantina(Time, Cantina, Temperatura, Umidita);

Time, Cantina → Temperatura, Umidita

Poiché gli attributi a sinistra sono superchiave, StatoCantina è in BCNF, quindi non necessita di ulteriori cambiamenti.

Lotto(Codice, Data, Durata, DataScadenza, Laboratorio, Dipendente);

Codice → Data, Durata, DataScadenza, Laboratorio, Dipendente

Poiché l'attributo a sinistra è superchiave, Lotto è in BCNF, quindi non necessita di ulteriori cambiamenti.

FaseLotto(Lotto, NumFase, Durata, Temperatura);

Lotto, NumFase → Durata, Temperatura

Poiché gli attributi a sinistra sono superchiave, FaseLotto è in BCNF, quindi non necessita di ulteriori cambiamenti.

Parametro(Nome);

Poiché non esistono dipendenze funzionali, Parametro è in BCNF.

Mungitrice(Codice, Modello, Marca);

Codice → Modello

Poiché l'attributo a sinistra è superchiave, Mungitrice è in BCNF, quindi non necessita di ulteriori cambiamenti.

Mungitura(Mungitrice, Time);

Poiché non esistono dipendenze funzionali, Mungitura è in BCNF.

Latte(Mungitrice, Time, Animale, pH, Densita, Glucosio, Caseina, Lattoalbumina, Temperatura, StatoSilos);

Mungitrice, Time, Animale → pH, Densita, Glucosio, Caseina, Lattoalbumina, Temperatura, StatoSilos

Poiché gli attributi a sinistra sono superchiave, Latte è in BCNF, quindi non necessita di ulteriori cambiamenti.

Ricetta(Prodotto, Descrizione, DataStagionatura);

Prodotto → Descrizione, DataStagionatura

Poiché l'attributo a sinistra è superchiave, Ricetta è in BCNF, quindi non necessita di ulteriori cambiamenti.

Fase(Prodotto, IDFase, Durata);

Prodotto, IDFase → Durata

Poiché gli attributi a sinistra sono superchiave, Fase è in BCNF, quindi non necessita di ulteriori cambiamenti.

Laboratorio(Codice, Tipo, Agriturismo);

Codice → Tipo, Agriturismo

Poiché l'attributo a sinistra è superchiave, Laboratorio è in BCNF, quindi non necessita di ulteriori cambiamenti.

Prelievo(Lotto, StatoSilos, Quantita);

Lotto, StatoSilos → Quantita

Poiché gli attributi a sinistra sono superchiave, Prelievo è in BCNF, quindi non necessita di ulteriori cambiamenti.

Avere(Lotto, NumFase, Parametro, Quantita);

Lotto, NumFase, Parametro → Quantita

Poiché gli attributi a sinistra sono superchiave, Avere è in BCNF, quindi non necessita di ulteriori cambiamenti.

Composta(Prodotto, IdFase, Parametro, Quantita);

Prodotto, IdFase, Parametro → Quantita

Poiché gli attributi a sinistra sono superchiave, Avere è in BCNF, quindi non necessita di ulteriori cambiamenti.

Impiegato(Lotto, Dipendente);

Poiché non esistono dipendenze funzionali, Impiegato è in BCNF.

- Area Soggiorno

Prenotazione(ID, DataArrivo, DataPartenza, Anticipo, UtenteRegistrato, UtenteNonRegistrato);

ID → DataArrivo, DataPartenza, Anticipo, UtenteRegistrato, UtenteNonRegistrato

Poiché l'attributo a sinistra è superchiave, Prenotazione è BCFN, quindi non necessita di ulteriori cambiamenti

Escursione(Nome, Costo, Itinerario, OraFine, OraInizio, GiornoSettimana, Guida);

Nome → Costo, Itinerario, OraFine, OraInizio, GiornoSettimana, Guida

Poiché l'attributo a sinistra è superchiave, Escursione è BCFN, quindi non necessita di ulteriori cambiamenti

Servizio(Nome, CostoUnitario);

Nome → CostoUnitario

Poiché l'attributo a sinistra è superchiave, Servizio è BCFN, quindi non necessita di ulteriori cambiamenti

Itinerario(Codice);

Poiché non esistono dipendenze funzionali, Itinerario è in BCNF.

Area(Itinerario, Ora, Nome, DurataSosta, Agriturismo);

Itinerario, Ora, Nome → DurataSosta, Agriturismo

Poiché gli attributi a sinistra sono superchiave, Area è BCFN, quindi non necessita di ulteriori cambiamenti

ServizioAggiuntivo(Servizio, TimeInizio, TimeFine, Prenotazione);

Poiché non esistono dipendenze funzionali, ServizioAggiuntivo è in BCNF.

Stanza(NumStanza, CostoStanza, Tipo);

NumStanza → CostoStanza, Tipo

Poiché l'attributo a sinistra è superchiave, Stanza è BCFN, quindi non necessita di ulteriori cambiamenti

Letto(CodLetto, TipoLetto, Stanza);

CodLetto → TipoLetto, Stanza

Poiché l'attributo a sinistra è superchiave, Letto è BCFN, quindi non necessita di ulteriori cambiamenti

Pagamento(Prenotazione, Time, Importo);

Prenotazione, Time → Importo

Poiché gli attributi a sinistra sono superchiave, Pagamento è BCFN, quindi non necessita di ulteriori cambiamenti

Registrato(Codice, Telefono, Nome, Cognome, CodCarta, Tipologia, NumDocumento, Scadenza, Ente);

Codice → Telefono, CodCarta, Tipologia, NumDocumento

NumDocumento → Nome, Cognome, Scadenza, Ente

Poiché nella seconda dipendenza l'attributo non è superchiave, bisognerà decomporre la tabella per ottenere BCNF.

Registrato(Codice, NumDocumento Telefono, CodCarta, Tipologia);

Documento(NumDocumento, Nome, Cognome, Scadenza, Ente);

Non occorre creare una nuova relazione poiché la chiave Codice è già contenuta in Registrato.

NonRegistrato(Codice, CodiceCarta);

Codice → CodiceCarta

Poiché l'attributo a sinistra è superchiave, NonRegistrato è BCFN, quindi non necessita di ulteriori cambiamenti

Riferimento(Prenotazione, Stanza);

Poiché non esistono dipendenze funzionali, Riferimento è in BCNF.

Associare(Prenotazione, Escursione, DataEscursione);

Prenotazione, Escursione → DataEscursione

Poiché l'attributo a sinistra è superchiave, Associare è BCNF, quindi non necessita di ulteriori cambiamenti

- Area Store

Account(NomeUtente, Password, DataIscrizione, Domanda, Risposta);

NomeUtente → Password, DataIscrizione, Domanda, Risposta

Poiché l'attributo a sinistra è superchiave, Account è BCNF, quindi non necessita di ulteriori cambiamenti

Ordine(Codice, Stato, Time, Account);

Codice → Stato, Time, Account

Poiché l'attributo a sinistra è superchiave, Ordine è BCNF, quindi non necessita di ulteriori cambiamenti

Recensione(Codice, Testo, Conservazione, Gradimento, Qualità, Gusto, UnitaProdotto, Data);

Codice → Testo, Conservazione, Gradimento, Qualità, Gusto, UnitaProdotto, Data

Poiché l'attributo a sinistra è superchiave, Recensione è BCNF, quindi non necessita di ulteriori cambiamenti

Reso(Codice, DataReso, UnitaProdotto);

Codice → DataReso, UnitaProdotto

Poiché l'attributo a sinistra è superchiave, Reso è BCNF, quindi non necessita di ulteriori cambiamenti

Spedizione(Codice, Stato);

Codice → Stato

Poiché l'attributo a sinistra è superchiave, Spedizione è BCNF, quindi non necessita di ulteriori cambiamenti

ParametriQualita (Nome);

Poiché non esistono dipendenze funzionali, ParametriQualita è in BCNF.

Hub(Codice, Citta, Indirizzo);

Codice → Citta, Indirizzo

Poiché l'attributo a sinistra è superchiave, Hub è BCFN, quindi non necessita di ulteriori cambiamenti

Trasporto(UnitaProdotto, Spedizione, DataPrevista, DataEffettiva);

UnitaProdotto, Spedizione → DataPrevista, DataEffettiva

Poiché gli attributi a sinistra sono superchiave, Trasporto è BCFN, quindi non necessita di ulteriori cambiamenti

Controllo(Reso, ParametroQualita, Valutazione);

Reso, ParametroQualita → Valutazione

Poiché gli attributi a sinistra sono superchiave, Controllo è BCFN, quindi non necessita di ulteriori cambiamenti

Prevedere(Spedizione, Hub, Time);

Spedizione, Hub → Time

Poiché gli attributi a sinistra sono superchiave, Prevedere è BCFN, quindi non necessita di ulteriori cambiamenti

Selezione(ProdottoCaseario, Ordine, Costo);

ProdottoCaseario, Ordine → Costo

Poiché gli attributi a sinistra sono superchiave, Selezione è BCFN, quindi non necessita di ulteriori cambiamenti

11. Trigger gestione attributi derivati

11.1 Calcolo del numero massimo di animali di ogni locale

Il risultato NumMassimo dipende dalla specie, quindi come già elencato nel calcolo di questo attributo derivato, si sono scelte delle dimensioni standard per ogni specie. L'obiettivo del trigger è quello di settare NumMassimo prima che l'intero record venga inserito andando a controllare quale sarà la specie dell'animale che sta per essere inserito.

```
DROP TRIGGER IF EXISTS triggerNumMassimo;
DELIMITER $$
CREATE TRIGGER triggerNumMassimo BEFORE INSERT ON locale
FOR EACH ROW
BEGIN
    DECLARE lung float default 0;
    DECLARE larg float default 0;

    IF NEW.Specie = "Bovini" THEN
        SET lung = 180;
        SET larg = 45;
    ELSEIF NEW.Specie = "Ovini" THEN
        SET lung = 140;
        SET larg = 35;
    ELSEIF NEW.Specie = "Caprini" THEN
        SET lung = 85;
        SET larg = 30;
    END IF;

    SET NEW.NumMassimo = (NEW.Lunghezza*NEW.Larghezza*10000)/(lung * larg);
END $$
```

11.2 Calcolo degli attributi Proteine, Glucidi, Fibre e Kcal

Si inseriscono i seguenti valori: “Peso del foraggio”, “Piante”, “Cereali”, “Frutta” (quest’ultimi tre, rappresentano la percentuale) . Questi ci permettono di calcolare il valore delle proteine, dei glucidi e delle fibre, e successivamente dopo aver calcolato tutti gli attributi, si procede al calcolo del KCal del foraggio.

```
DROP TRIGGER IF EXISTS triggerProteine_Glucidi_Fibre_KCal;
DELIMITER $$
CREATE TRIGGER triggerProteine_Glucidi_Fibre_KCal BEFORE INSERT ON foraggio
FOR EACH ROW
BEGIN
    DECLARE pesoGrammi float default 0;
    DECLARE pesoPiante float default 0;
    DECLARE pesoCereali float default 0;
    DECLARE pesoFrutta float default 0;
    DECLARE proteine float default 0;
    DECLARE glucidi float default 0;
    DECLARE fibre float default 0;

    -- Conversione da chili in grammi
    SET pesoGrammi = NEW.Peso * 1000;

    -- calcolo peso dei singoli componenti
    SET pesoPiante = pesoGrammi * NEW.Piante / 100;
    SET pesoCereali = pesoGrammi * NEW.Cereali / 100;
    SET pesoFrutta = pesoGrammi * NEW.Frutta / 100;

    - calcola e inserisce nel record la quantità di proteine, glucidi e fibre
    presenti
    SET proteine = (pesoCereali*11.5)+(pesoPiante *2.7)+(pesoFrutta*1.2);
    SET glucidi = (pesoCereali*75)+(pesoPiante*3.2)+(pesoFrutta*16);
    SET fibre = (pesoCereali*2.5)+(pesoPiante*3)+(pesoFrutta*2.8);

    SET NEW.Proteine = proteine;
    SET NEW.Glucidi = glucidi;
    SET NEW.Fibre = fibre;

    -- calcolo delle KCal totali
    SET NEW.KCal = ((proteine * 4.0)+(glucidi * 3.8)+(fibre * 2.0))/ 1000;

END $$
```

11.3 Gestione dell'inserimento/decremento del latte sul silos

Questo trigger ha la funzione di aggiungere il latte all'interno dei silos. Come prima cosa viene scelto uno dei silos tra quelli disponibili seguendo il seguente criterio:

- La composizione chimico-fisica deve essere simile
- Se la prima condizione non si verifica, si sceglie un silos vuoto
- Se neanche la seconda condizione può essere soddisfatta, l'utente viene avvisato con un messaggio d'errore

Per il decremento basta sottrarre la quantità di latte prelevata dal silos.

```
DROP TRIGGER IF EXISTS triggerInserimentoSulSilos;
DELIMITER $$
CREATE TRIGGER triggerInserimentoSulSilos BEFORE INSERT ON latte
FOR EACH ROW
BEGIN
    DECLARE silosSimile int default 0;
    DECLARE ph float default 0;
    DECLARE densita float default 0;
    DECLARE glucosio float default 0;
    DECLARE caseina float default 0;
    DECLARE lattoalbumina float default 0;
    DECLARE temperatura float default 0;

    -- Ricerca del silos target
    SELECT SS.Codice, SS.pH, SS.Densita, SS.Glucosio, SS.Caseina,
    SS.Lattoalbumina, SS.Temperatura
    INTO silosSimile, ph, densita, glucosio, caseina, lattoalbumina, temperatura
    FROM statosilos SS
    WHERE (SS.LivRiempimento + NEW.Quantita < SS.Capacita)
    AND
    (
        (NEW.pH BETWEEN (SS.pH - SS.pH * 0.2) AND (SS.pH + SS.pH * 0.2))
        AND
        (NEW.Densita BETWEEN (SS.Densita - SS.Densita * 0.2) AND
        (SS.Densita + SS.Densita * 0.2))
        AND
        (NEW.Glucosio BETWEEN (SS.Glucosio - SS.Glucosio * 0.2) AND
        (SS.Glucosio + SS.Glucosio * 0.2))
        AND
        (NEW.Caseina BETWEEN (SS.Caseina - SS.Caseina * 0.2) AND
        (SS.Caseina + SS.Caseina * 0.2))
        AND
        (NEW.Lattoalbumina BETWEEN (SS.Lattoalbumina-SS.Lattoalbumina* 0.2)
        AND (SS.Lattoalbumina + SS.Lattoalbumina * 0.2))
        AND
        (NEW.Temperatura BETWEEN (SS.Temperatura - SS.Temperatura * 0.2)
        AND (SS.Temperatura + SS.Temperatura * 0.2))
    OR
    SS.LivRiempimento = 0
    )
    AND silosSimile = 0;
END
```

```

        )
        LIMIT 1;
IF silosSimile = 0 THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = "Non c'è spazio disponibile nei silos per inserire il latte";
END IF;

- Ogni nuovo valore sarà dato dalla media tra il valore dello stato silos e
  il nuovo valore del latte
IF ph = 0 THEN SET ph = NEW.ph;
ELSE          SET ph = (ph + NEW.ph) / 2;
END IF;

IF densita = 0 THEN SET densita = NEW.densita;
ELSE               SET densita = (densita + NEW.densita) / 2;
END IF;

IF glucosio = 0 THEN SET glucosio = NEW.glucosio;
ELSE                SET glucosio = (glucosio + NEW.glucosio) / 2;
END IF;

IF caseina = 0 THEN SET caseina = NEW.caseina;
ELSE               SET caseina = (caseina + NEW.caseina) / 2;
END IF;

IF lattoalbumina = 0 THEN SET lattoalbumina = NEW.lattoalbumina;
ELSE                     SET lattoalbumina = (lattoalbumina + NEW.lattoalbumina) / 2;
END IF;

IF temperatura = 0 THEN SET temperatura = NEW.temperatura;
ELSE                   SET temperatura = (temperatura + NEW.temperatura) / 2;
END IF;

-- Aggiorna la variabile LivRiempimento del silos target
UPDATE statosilos SS
SET SS.LivRiempimento = SS.LivRiempimento + NEW.Quantita,
    SS.ph = ph,
    SS.densita = densita,
    SS.glucosio = glucosio,
    SS.caseina = caseina,
    SS.lattoalbumina = lattoalbumina,
    SS.temperatura = temperatura
WHERE SS.Codice = silosSimile;

-
- Settaggio dell'attributo StatoSilos, cioè il silos dove il latte è stato inserito
  SET NEW.StatoSilos = silosSimile;

END $$

```

```
DROP TRIGGER IF EXISTS triggerDecrementoSulSilos;
DELIMITER $$
CREATE TRIGGER triggerDecrementoSulSilos AFTER INSERT ON prelievo
FOR EACH ROW
BEGIN

    -- Aggiorna la variabile LivRiempimento del silos target
    UPDATE statosilos
    SET LivRiempimento = IF(LivRiempimento - NEW.Quantita < 0, 0,
LivRiempimento - NEW.Quantita)
    WHERE NEW.StatoSilos = Codice;

END $$
```

11.4 Gestione del pagamento

Calcolo dell'importo da pagare, seguendo la modalità descritta precedentemente nella sezione degli attributi derivata.

```
DROP TRIGGER IF EXISTS trigger_Pagamento;
DELIMITER $$
CREATE TRIGGER trigger_Pagamento
BEFORE INSERT ON pagamento
FOR EACH ROW
BEGIN
    DECLARE CostoStanze FLOAT DEFAULT 0;
    DECLARE DataPartenza DATE;
    DECLARE DataArrivo DATE;
    DECLARE Anticipo FLOAT DEFAULT 0;
    DECLARE CostoEsc FLOAT DEFAULT 0;
    DECLARE CostoServizi FLOAT DEFAULT 0;

    -- PARTE STANZE
    SET DataPartenza = (
        SELECT DataPartenza
        FROM prenotazione
        WHERE ID = NEW.Prenotazione
    );

    SET DataArrivo =(
        SELECT DataArrivo
        FROM prenotazione
        WHERE ID = NEW.Prenotazione
    );

    SET Anticipo = (
        SELECT Anticipo
        FROM prenotazione
        WHERE ID = NEW.Prenotazione
    );

    -- ricavo il costo totale delle stanze
    SET CostoStanze = (
        SELECT SUM(S.CostoStanza)
        FROM Riferimento R
            INNER JOIN
                Stanza S ON NumStanza = Stanza
        WHERE R.Prenotazione = NEW.Prenotazione
    );

    -- PARTE ESCURSIONI
    SELECT SUM(E.Costo)
    FROM associare A
        INNER JOIN
```



```

        escursione E ON E.Nome = A.Escursione
WHERE A.Prenotazione = NEW.Prenotazione
INTO CostoEsc;

-- PARTE SERVIZI AGGIUNTIVI

SELECT SUM(S.CostoUnitario*(hour(TIMEDIFF(SA.TimeFine, SA.TimeInizio))))
FROM servizioaggiuntivo SA
    INNER JOIN
        servizio S ON S.Nome = SA.Servizio
WHERE SA.Prenotazione = NEW.Prenotazione
INTO CostoServizi;

-- inserimento dell'importo del pagamento
SET NEW.Importo = CostoStanze * (DATEDIFF(DataPartenza, DataArrivo))
                + CostoEsc + CostoServizi
                - Anticipo;

END $$

```

12. Business Rule

12.1 Gestione stato mangiatoie:

Questo event serve per riempire le mangiatoie. Tutte le mangiatoie verranno prese in considerazione, e per ognuna si sceglie un foraggio la cui quantità sarà pari a 100.

```
DROP EVENT IF EXISTS ImmettiPasto;
DELIMITER $$
CREATE EVENT ImmettiPasto
ON SCHEDULE EVERY 12 HOUR
STARTS '2020-01-16 12:00:00'
DO
BEGIN
    DECLARE mangiatoia INT DEFAULT 0;
    DECLARE finito INT DEFAULT 0;
    DECLARE randForaggio INT DEFAULT 0;

    DECLARE cursore_mangiatoie CURSOR FOR
        SELECT Codice
        FROM mangiatoia;

    DECLARE CONTINUE HANDLER FOR NOT FOUND
    SET finito = 1;

    OPEN cursore_mangiatoie;
    WHILE finito = 0 DO
        FETCH cursore_mangiatoie INTO mangiatoia;

        SELECT ROUND(RAND() * (SELECT MAX(Codice) FROM foraggio))
        INTO randForaggio;

        INSERT INTO statomangiatoia
        VALUES(mangiatoia, current_timestamp, 100, randForaggio);

    END WHILE;

    CLOSE cursore_mangiatoie;
END $$
```

L'attivazione di questo event, provoca il reset delle mangiatoie, ovvero il foraggio al loro interno viene scartato visto che sono trascorse 6 ore da quando è stato inserito.

```
DROP EVENT IF EXISTS ResetPastoPranzo;
DELIMITER $$
CREATE EVENT ResetPastoPranzo
ON SCHEDULE EVERY 1 DAY
STARTS '2020-01-16 18:00:00'
DO
BEGIN
    DECLARE mangiatoia INT DEFAULT 0;
    DECLARE finito INT DEFAULT 0;

    DECLARE cursore_mangiatoie CURSOR FOR
        SELECT Codice
        FROM mangiatoia;
    DECLARE CONTINUE HANDLER FOR NOT FOUND
    SET finito = 1;

    OPEN cursore_mangiatoie;
    WHILE finito = 0 DO
        FETCH cursore_mangiatoie INTO mangiatoia;

        INSERT INTO statomangiatoia
        VALUES(mangiatoia, current_timestamp, 0, null);

    END WHILE;
    CLOSE cursore_mangiatoie;

END $$
```

L'attivazione di questo event, provoca il reset delle mangiatoie, ovvero il foraggio al loro interno viene scartato visto che sono trascorse 6 ore da quando è stato inserito.

```
DROP EVENT IF EXISTS ResetPastoCena;
DELIMITER $$
CREATE EVENT ResetPastoCena
ON SCHEDULE EVERY 1 DAY
STARTS '2020-01-16 01:00:00'
DO
BEGIN
    DECLARE mangiatoia INT DEFAULT 0;
    DECLARE finito INT DEFAULT 0;

    DECLARE cursore_mangiatoie CURSOR FOR
        SELECT Codice
        FROM mangiatoia;
    DECLARE CONTINUE HANDLER FOR NOT FOUND
    SET finito = 1;

    OPEN cursore_mangiatoie;
    WHILE finito = 0 DO
        FETCH cursore_mangiatoie INTO mangiatoia;

        INSERT INTO statomangiatoia
        VALUES(mangiatoia, current_timestamp, 0, null);

    END WHILE;
    CLOSE cursore_mangiatoie;
END $$
```

Questo event viene eseguito 30 minuti. La sua funzionalità è quella di decrementare la quantità di foraggio di tutte le mangiatoie del 20%.

```
DROP EVENT IF EXISTS CheckPasto;
DELIMITER $$
CREATE EVENT CheckPasto
ON SCHEDULE EVERY 30 MINUTE
STARTS '2020-01-16 12:30:00'
DO
BEGIN
    DECLARE mangiatoia INT DEFAULT 0;
    DECLARE finito INT DEFAULT 0;
    DECLARE lastForaggio INT DEFAULT 0;
    DECLARE lastQuantita INT DEFAULT 0;
    DECLARE quantitaDaInserire INT DEFAULT 0;

    DECLARE cursore_mangiatoie CURSOR FOR
        SELECT Codice
        FROM mangiatoia;
    DECLARE CONTINUE HANDLER FOR NOT FOUND
    SET finito = 1;

    OPEN cursore_mangiatoie;
    WHILE finito = 0 DO
        FETCH cursore_mangiatoie INTO mangiatoia;

        -- query per trovare il foraggio e la quantita usato nell'ultimo check
        SELECT SM.Foraggio, SM.Quantita
        FROM statomangiatoia SM
        WHERE SM.Istante = (
            SELECT MAX(SM2.Istante)
            FROM statomangiatoia SM2
            WHERE SM2.Mangiatoia = mangiatoia
        )
        AND SM.Mangiatoia = mangiatoia
        INTO lastForaggio, lastQuantita;

        SET quantitaDaInserire = (
            IF(lastQuantita - lastQuantita*0.2 < 0, 0,
                lastQuantita - lastQuantita*0.2)
        );

        IF lastForaggio IS NOT NULL THEN
            INSERT INTO statomangiatoia
            VALUES(mangiatoia, current_timestamp, quantitaDaInserire, lastForaggio);
        END IF;

    END WHILE;
    CLOSE cursore_mangiatoie;
END $$
```

12.2 Trigger per l'inserimento della tipologia del foraggio

```
DROP TRIGGER IF EXISTS check_typeForaggio;
DELIMITER $$
CREATE TRIGGER check_typeForaggio BEFORE INSERT ON foraggio
FOR EACH ROW
BEGIN

    IF NEW.Tipo <> 'F' AND NEW.Tipo <> 'C' THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Tipologia foraggio non valida!";
    END IF;

END $$
```

12.3 Trigger per l'inserimento della tipologia del dipendente

```
DROP TRIGGER IF EXISTS check_typeDipendente;
DELIMITER $$
CREATE TRIGGER check_typeDipendente BEFORE INSERT ON dipendente
FOR EACH ROW
BEGIN

    IF NEW.Tipologia <> 'L' AND NEW.Tipologia <> 'G' AND NEW.Tipologia <> 'P' THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Tipologia dipendente non valida!";
    END IF;

END $$
```

12.4 Trigger per la tipologia di CondizionamentoIlluminazione

```
DROP TRIGGER IF EXISTS check_typeCondizIllum;
DELIMITER $$
CREATE TRIGGER check_typeCondizIllum BEFORE INSERT ON condizionamentoilluminazione
FOR EACH ROW
BEGIN

    IF NEW.Tipo <> 'I' AND NEW.Tipo <> 'C' THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Tipologia non valida!";
    END IF;

END $$
```

12.5 Trigger per l'inserimento dello stato di riproduzione

```
DROP TRIGGER IF EXISTS check_statoRiproduzione;
DELIMITER $$
CREATE TRIGGER check_statoRiproduzione BEFORE INSERT ON riproduzione
FOR EACH ROW
BEGIN

    IF NEW.Stato <> 'Successo' AND NEW.Stato <> 'Insuccesso' THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Stato non valido!";
    END IF;

END $$
```

12.6 Gestione prenotazione servizi aggiuntivi

Questo trigger è stato implementato per gestire la richiesta dei servizi aggiuntivi. I clienti che possono usufruire di tali servizi, sono solo quelli che hanno prenotato almeno una camera "Suite". Occorre quindi effettuare un controllo prima di poter effettuare le prenotazioni.

```
-- Controllo sui servizi aggiuntivi se il cliente aveva prenotato una suite
DROP TRIGGER IF EXISTS triggerServizioAggiuntivo;
DELIMITER $$
CREATE TRIGGER triggerServizioAggiuntivo BEFORE INSERT ON servizioaggiuntivo
FOR EACH ROW
BEGIN
    DECLARE isSuite INT DEFAULT 0;

    -- conta il numero di suite prenotate, se > 0 allora va bene
    SELECT COUNT(*)
    FROM   riferimento R
          INNER JOIN
            stanza S ON R.Stanza = S.NumStanza
    WHERE  R.Prenotazione = NEW.Prenotazione
          AND S.Tipo = 'Suite'
    INTO isSuite;

    IF isSuite = 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Non è possibile aggiungere servizi aggiuntivi!';
    END IF;

END $$
```

12.7 Gestione del pascolo

Questo trigger consente di registrare su una tabella, le posizioni iniziali di tutti gli animali dello stesso locale durante il pascolo.

```
DROP TRIGGER IF EXISTS trigger_pascolo;
DELIMITER $$
CREATE TRIGGER trigger_pascolo AFTER INSERT ON pascolo
FOR EACH ROW
BEGIN

    DECLARE MediaLongitudiniRecintiZona FLOAT DEFAULT 0; -- X
    DECLARE MediaLatitudiniRecintiZona FLOAT DEFAULT 0; -- Y

    -- inserimento dentro la tabella 'andare' la coppia:
    -- codice pascolo appena aggiunto e animali appartenenti a quel locale
    INSERT INTO andare
        SELECT A.Codice, NEW.Codice
        FROM animale A
        WHERE A.CodiceLocale = NEW.Locale;

    -- solo di un punto perchè tanto l'altro punto è ridondante nella somma
    SELECT AVG(X1), AVG(Y1)
    FROM recinzione R
    WHERE R.NomeZona = NEW.NomeZona
    INTO MediaLongitudiniRecintiZona, MediaLatitudiniRecintiZona;

    -- con una variazione di +- 2
    INSERT INTO gps(Longitudine, Latitudine, Time, Animale)
        SELECT (MediaLongitudiniRecintiZona + RAND() * (-2) + RAND() * (2)),
            (MediaLatitudiniRecintiZona + RAND() * (-2) + RAND() * (2)),
            current_timestamp(),
            A.Codice
        FROM animale A
        WHERE A.CodiceLocale = NEW.Locale;

END $$
```


12.8 Gestione inserimento delle fasce d'orario

```
DROP TRIGGER IF EXISTS check_orario;
DELIMITER $$
CREATE TRIGGER check_orario BEFORE INSERT ON orario
FOR EACH ROW
BEGIN

    IF NEW.Ora <> 'M' AND NEW.Ora <> 'P' AND NEW.Ora <> 'S' THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Ora non valida!";
    END IF;

END $$
```

12.9 Gestione della prenotazione delle escursioni

```
DROP TRIGGER IF EXISTS trigger_Escursioni;
DELIMITER $$
CREATE TRIGGER trigger_Escursioni BEFORE INSERT ON associare
FOR EACH ROW
BEGIN
    IF NEW.DataPrenotazione < CURRENT_DATE + INTERVAL 2 DAY THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "L'escursione deve essere prenotata con 48 ore
                                di anticipo";
    END IF;
END $$
```

12.10 Gestione dello stato della cantina

Questo event viene eseguito una volta al giorno e ha la funzionalità di aggiornare lo stato di ogni cantina. In particolare vengono salvate nel database le nuove misurazioni della temperatura e dell'umidità.

```
DROP EVENT IF EXISTS event_statoCantina;
DELIMITER $$
CREATE EVENT event_statoCantina
ON SCHEDULE EVERY 1 DAY
STARTS '2020-01-14 12:00:00'
DO
BEGIN
    DECLARE cantina INT DEFAULT 0;
    DECLARE finito INT DEFAULT 0;
    DECLARE randTemp FLOAT DEFAULT 0;
    DECLARE randUmi FLOAT DEFAULT 0;

    DECLARE cursore_cantina CURSOR FOR
        SELECT Codice
        FROM cantina;

    DECLARE CONTINUE HANDLER FOR NOT FOUND
    SET finito = 1;

    OPEN cursore_cantina;
    WHILE finito = 0 DO
        FETCH cursore_cantina INTO cantina;

        SET randTemp = ROUND(5 + RAND() * 20);

        SET randUmi = ROUND(40 + RAND() * 100);

        INSERT INTO statocantina
        VALUES(current_timestamp, cantina, randTemp, randUmi);

    END WHILE;

    CLOSE cursore_cantina;
END $$
```

12.11 Gestione dello stato degli abbeveratoi

Questo event è stato creato per cambiare ogni giorno, la quantità di vitamine e di sali minerali disciolte nell'acqua.

```
DROP EVENT IF EXISTS CheckWater;
DELIMITER $$
CREATE EVENT CheckWater
ON SCHEDULE EVERY 1 DAY
STARTS '2020-01-16 06:00:00'
DO
BEGIN
    DECLARE abbeveratoio INT DEFAULT 0;
    DECLARE finito INT DEFAULT 0;
    DECLARE quantitaSali INT DEFAULT 0;
    DECLARE quantitaVitamine INT DEFAULT 0;

    DECLARE cursore_abbeveratoio CURSOR FOR
        SELECT Codice
        FROM abbeveratoio;
    DECLARE CONTINUE HANDLER FOR NOT FOUND
    SET finito = 1;

    OPEN cursore_abbeveratoio;
    WHILE finito = 0 DO
        FETCH cursore_abbeveratoio INTO abbeveratoio;

        SET quantitaSali = (
            SELECT ROUND(RAND() * 10)
        );

        SET quantitaVitamine = (
            SELECT ROUND(RAND() * 10)
        );

        INSERT INTO statoabbeveratoio
        VALUES(abbeveratoio, current_timestamp, quantitaSali, quantitaVitamine, 100);

    END WHILE;
    CLOSE cursore_abbeveratoio;
END $$
```

Questo event, invocato ogni ora, serve per decrementare la quantità di acqua presente in tutti gli abbeveratoi, e per reinserirla quando scende sotto una soglia limite.

```
DROP EVENT IF EXISTS updateWater;
DELIMITER $$
CREATE EVENT updateWater
ON SCHEDULE EVERY 1 HOUR
STARTS '2020-01-16 06:30:00'
DO
BEGIN
    DECLARE abbeveratoio INT DEFAULT 0;
    DECLARE finito INT DEFAULT 0;
    DECLARE lastSali INT DEFAULT 0;
    DECLARE lastVitamine INT DEFAULT 0;
    DECLARE lastQuantita INT DEFAULT 0;
    DECLARE quantitaDaInserire INT DEFAULT 0;

    DECLARE cursore_abbeveratoio CURSOR FOR
        SELECT Codice
        FROM abbeveratoio;

    DECLARE CONTINUE HANDLER FOR NOT FOUND
    SET finito = 1;

    OPEN cursore_abbeveratoio;
    WHILE finito = 0 DO
        FETCH cursore_abbeveratoio INTO abbeveratoio;

        -- query per trovare i sali minerali, vitamine e
        -- quantità usati precedentemente

        SELECT SA.SaliMinaerali, SA.Vitamine, SA.Quantita
        FROM statoabbeveratoio SA
        WHERE SA.Istante = (
            SELECT MAX(SA2.Istante)
            FROM statoabbeveratoio SA2
            WHERE SA2.Abbeveratoio = abbeveratoio
        )
        AND SA.Abbeveratoio = abbeveratoio
        INTO lastSali, lastVitamine, lastQuantita;

        -- quando l'acqua è minore o uguale al 20% della sua quantità,
        -- viene risettata a 100

        SET quantitaDaInserire = (
            IF(lastQuantita - lastQuantita*0.2 <= lastQuantita*0.2 , 100,
                lastQuantita - lastQuantita*0.2)
        );
    END WHILE;
END;
```

```

INSERT INTO statoabbeveratoio
VALUES(abbeveratoio, current_timestamp, lastSali, lastVitamine,
                                             quantitaDaInserire);

END WHILE;
CLOSE cursore_abbeveratoio;

END $$

```

12.12 Gestione dell'aggiornamento del gps

Questo event aggiorna ogni 10 secondi, la posizione di ogni animale, facendola variare rispetto alla precedente di +/- 0.5.

```

DROP EVENT IF EXISTS event_updateGPS;
DELIMITER $$
CREATE EVENT event_updateGPS
ON SCHEDULE EVERY 10 SECOND
DO
BEGIN
    -- inserimento animali con posizione aggiornata
    INSERT INTO gps(Longitudine, Latitudine, Time, Animale)
    SELECT  G.Longitudine + RAND() * (-0.5) + RAND() * (0.5),
            G.Latitudine + RAND() * (-0.5) + RAND() * (0.5),
            CURRENT_TIMESTAMP(),
            G.Animale
    FROM    gps G
    WHERE   G.Time >= ALL ( -- andiamo a vedere l'ultima sua posizione
                            SELECT  G2.Time
                            FROM    gps G2
                            WHERE   G2.Animale = G.Animale
                            )
            AND G.Animale IS NOT NULL
    GROUP BY
            G.Animale;
END $$

```

12.13 Gestione inserimento della tipologia di stanza

```
DROP TRIGGER IF EXISTS check_typeStanza;
DELIMITER $$
CREATE TRIGGER check_typeStanza BEFORE INSERT ON stanza
FOR EACH ROW
BEGIN

    IF NEW.Tipo <> 'Semplice' AND NEW.Tipo <> 'Suite' THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Tipologia stanza non valida!";
    END IF;

END $$
```

12.14 Gestione inserimento dello stato dell'ordine

```
DROP TRIGGER IF EXISTS check_statoOrdine;
DELIMITER $$
CREATE TRIGGER check_statoOrdine BEFORE INSERT ON ordine
FOR EACH ROW
BEGIN

    IF NEW.Stato <> 'Pendente' AND NEW.Stato <> 'Processazione' AND
        NEW.Stato <> 'Evaso' AND NEW.Stato <> 'In Preparazione' AND
        NEW.Stato <> 'Spedito' THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Stato ordine non valido!";
    END IF;

END $$
```

12.15 Gestione dell'inserimento dei parametri della recensione

```
DROP TRIGGER IF EXISTS check_recensione;
DELIMITER $$
CREATE TRIGGER check_recensione BEFORE INSERT ON recensione
FOR EACH ROW
BEGIN

    IF NEW.Gradimento < 1 OR NEW.Gradimento > 10 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Inserimento Gradimento non valido!";
    END IF;

    IF NEW.Qualita < 1 OR NEW.Qualita > 10 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Inserimento Qualita non valido!";
    END IF;

    IF NEW.Gusto < 1 OR NEW.Gusto > 10 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Inserimento Gusto non valido!";
    END IF;

END $$
```

12.16 Gestione dell'inserimento della tipologia di letto

```
DROP TRIGGER IF EXISTS check_typeLetto;
DELIMITER $$
CREATE TRIGGER check_typeLetto BEFORE INSERT ON letto
FOR EACH ROW
BEGIN

    IF NEW.TipoLetto <> 'S' AND NEW.TipoLetto <> 'M' THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Tipologia letto non valida!";
    END IF;

END $$
```

12.17 Gestione spedizione

Quest'event permette di aggiornare giornalmente lo stato della spedizione facendola avanzare di uno stato.

```
DROP EVENT IF EXISTS event_Spedizione;
DELIMITER $$
CREATE EVENT event_Spedizione
ON SCHEDULE EVERY 1 DAY
STARTS '2020-01-15 12:00:00'
DO
BEGIN
    DECLARE finito INTEGER DEFAULT 0;
    DECLARE statoCorrente VARCHAR(50) DEFAULT "";
    DECLARE statoSuccessivo VARCHAR(50) DEFAULT "";
    DECLARE Spedizione INTEGER DEFAULT 0;

    DECLARE cursoreSpedizioni CURSOR FOR
        SELECT S.Codice, S.Stato
        FROM Spedizione S
        WHERE S.Stato <> "Consegnato";

    DECLARE CONTINUE HANDLER FOR NOT FOUND
        SET finito = 1;

    OPEN cursoreSpedizioni;

    WHILE finito = 0 DO
        FETCH cursoreSpedizioni INTO Spedizione, statoCorrente;

        CASE
            WHEN statoCorrente = "Spedita" THEN
                SET statoSuccessivo = "In transito";
            WHEN statoCorrente = "In transito" THEN
                SET statoSuccessivo = "In consegna";
            WHEN statoCorrente = "In consegna" THEN
                SET statoSuccessivo = "Consegnata";
            UPDATE Trasporto T
            SET DataEffettiva = CURRENT_DATE;
            WHERE T.Spedizione = Spedizione;

        END CASE;

        UPDATE Spedizione S
        SET Stato = statoSuccessivo
        WHERE S.Codice = Spedizione;

    END WHILE;
    CLOSE cursoreSpedizioni;
END $$
```


12.18 Gestione dell'inserimento della tipologia di carta

```
DROP TRIGGER IF EXISTS check_typeCarta;
DELIMITER $$
CREATE TRIGGER check_typeCarta BEFORE INSERT ON registrato
FOR EACH ROW
BEGIN

    IF NEW.Tipologia <> 'CC' AND NEW.Tipologia <> 'PP' THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Tipologia carta non valida!";
    END IF;

END $$
```

12.19 Gestione dell'inserimento di un reso

```
DROP TRIGGER IF EXISTS trigger_Reso;
DELIMITER $$
CREATE TRIGGER trigger_Reso BEFORE INSERT ON Reso
FOR EACH ROW
BEGIN
    DECLARE dataConsegna date;

    SELECT DataEffettiva INTO dataConsegna
    FROM trasporto
    WHERE UnitaProdotto = NEW.UnitaProdotto;

    IF dataConsegna + INTERVAL 2 DAY < NEW.DataReso THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Attenzione, il reso deve essere effettuato entro 48 ore
                                dalla consegna!";

    END IF;
END $$
```

12.20 Gestione creazione nuove tabelle dopo l'inserimento su riproduzione

Questo trigger permette di controllare lo stato di una riproduzione. Se questa si conclude con successo, allora verranno inseriti due nuovi record rispettivamente uno su scheda di gestazione e l'altro su intervento di controllo, in caso contrario, non saranno effettuate modifiche.

```
DROP TRIGGER IF EXISTS trigger_Riproduzione;
DELIMITER $$
CREATE TRIGGER trigger_Riproduzione AFTER INSERT ON riproduzione
FOR EACH ROW
BEGIN
    DECLARE randVeterinario int default 0;

    IF NEW.Stato = "Successo" THEN
        -- Scelta casuale del veterinario
        WHILE randVeterinario = 0 DO
            SELECT ROUND(RAND() * (SELECT COUNT(*) FROM veterinario))
            INTO randVeterinario;
        END WHILE;

        -- creazione scheda gestazione animale
        INSERT INTO schedadigestazione
        VALUES (NEW.Codice, randVeterinario);

        -- Aggiunta di un intervento di controllo programmato
        INSERT INTO interventodicontrollo (DataProgrammata, DataControllo,
            Stato, Esito, Veterinario, CodiceRiproduzione, Terapia)
        VALUES (CURRENT_DATE + INTERVAL 3 MONTH, null, "Programmato", null,
            null, NEW.Codice, null);
    END IF;
END $$
```

12.21 Gestione dell'inserimento delle date della prenotazione

```
DROP TRIGGER IF EXISTS check_datePrenotazione;
DELIMITER $$
CREATE TRIGGER check_datePrenotazione BEFORE INSERT ON prenotazione
FOR EACH ROW
BEGIN
    IF NEW.DataPartenza < NEW.DataArrivo THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Inserimento date non valido!";
    END IF;
END $$
```

12.22 Gestione dell'aggiornamento dei parametri dei locali

Questo event, è stato creato per memorizzare le misurazioni dei sensori di ogni locale dell'agriturismo. Ciò che viene fatto è la generazione casuale di nuovi parametri e il loro successivo salvataggio.

```
DROP EVENT IF EXISTS event_parametriLocale;
DELIMITER $$
CREATE EVENT event_parametriLocale
ON SCHEDULE EVERY 4 HOUR DO
BEGIN
    DECLARE sensori INT DEFAULT 0;
    DECLARE temperatura INT DEFAULT 0;
    DECLARE umidita INT DEFAULT 0;
    DECLARE LivAzoto INT DEFAULT 0;
    DECLARE LivMetano INT DEFAULT 0;
    DECLARE LivSporco INT DEFAULT 0;
    DECLARE finito INT DEFAULT 0;

    DECLARE cursore CURSOR FOR
        SELECT Codice
        FROM sensore;
    DECLARE CONTINUE HANDLER FOR NOT FOUND
    SET finito = 1;

    OPEN cursore;
    WHILE finito = 0 DO
        FETCH cursore INTO sensori;

        SELECT ROUND(5 + RAND() * 30)
        INTO temperatura;
        SELECT ROUND(40 + RAND() * 60)
        INTO umidita;
        SELECT ROUND(RAND() * 10)
        INTO LivAzoto;
        SELECT ROUND(RAND() * 10)
        INTO LivMetano;
        SELECT ROUND(RAND() * 10)
        INTO LivSporco;

        UPDATE sensore S
        SET S.temperatura = temperatura,
            S.Umidita = umidita,
            S.LivAzoto = LivAzoto,
            S.LivSporco = LivSporco
        WHERE S.Codice = sensori;

    END WHILE;
    CLOSE cursore;
END $$
```

12.23 Gestione dell'inserimento del giorno della settimana delle escursioni

```
DROP TRIGGER IF EXISTS check_orariEscursione;
DELIMITER $$
CREATE TRIGGER check_orariEscursione BEFORE INSERT ON escursione
FOR EACH ROW
BEGIN

    IF NEW.OraFine < NEW.OraInizio THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Inserimento orari non valido!";
    END IF;

    IF NEW.GiornoSettimana <> 'LU' AND NEW.GiornoSettimana <> 'MA' AND
        NEW.GiornoSettimana <> 'ME' AND NEW.GiornoSettimana <> 'GI' AND
        NEW.GiornoSettimana <> 'VE' AND NEW.GiornoSettimana <> 'SA' AND
        NEW.GiornoSettimana <> 'DO' THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Giorno settimana non valido!";
    END IF;
END $$
```

12.24 Gestione delle pulizia dei locali

Questo trigger permette di gestire la pulizia dei locali, infatti ogni volta che viene effettuata una misurazione da parte dei sensori, viene controllato se uno dei parametri del locale supera una certa soglia. Se il controllo ha esito positivo, allora viene richiesta la pulizia del locale.

L'event che segue, è stato creato per gestire lo stato della pulizia. Infatti ogni 6 ore (quando si attiva), vengono selezionati tutti i locali per i quali è stato richiesto un intervento di pulizia e si procede con una pulizia "virtuale", azzerando quindi la richiesta e sostituendola con lo stato "Effettuato"

```
DROP TRIGGER IF EXISTS Trigger_StatoSensore;
DELIMITER $$
CREATE TRIGGER Trigger_StatoSensore BEFORE UPDATE ON sensore
FOR EACH ROW
BEGIN
    DECLARE sogliaSporcizia INTEGER DEFAULT 5;
    DECLARE sogliaAzoto INTEGER DEFAULT 5;
    DECLARE sogliaMetano INTEGER DEFAULT 5;
    DECLARE flag BOOLEAN DEFAULT FALSE;
    DECLARE LivSporco INTEGER DEFAULT 0;
    DECLARE LivAzoto INTEGER DEFAULT 0;
    DECLARE LivMetano INTEGER DEFAULT 0;

    DECLARE sensore INTEGER DEFAULT 0;

    INSERT INTO statosensore
    VALUES(current_timestamp(), NEW.Codice, null);

    SELECT S.LivAzoto, S.LivMetano, S.LivSporco INTO LivAzoto, LivMetano, LivSporco
    FROM Sensore S
    WHERE S.Codice = NEW.Codice;

    IF LivSporco > sogliaSporcizia THEN
        SET flag = TRUE;
    ELSEIF LivAzoto > sogliaAzoto THEN
        SET flag = TRUE;
    ELSEIF LivMetano > sogliaMetano THEN
        SET flag = TRUE;
    END IF;

    IF flag = TRUE THEN
        UPDATE statosensore
        SET stato = "Richiesto";
    END IF;

END $$
```

```

DROP EVENT IF EXISTS pulizia_effettuata;
DELIMITER $$
CREATE EVENT pulizia_effettuata
ON SCHEDULE EVERY 6 HOUR
STARTS '2020-01-13 12:00:00'
DO
BEGIN

    DECLARE time TIMESTAMP;
    DECLARE sensore INTEGER;
    DECLARE finito INTEGER DEFAULT 0;

    DECLARE cursore CURSOR FOR
        SELECT SS.Time, SS.Sensore
        FROM statosensore SS
        WHERE Stato = "Richiesto";

    DECLARE CONTINUE HANDLER FOR NOT FOUND
        SET finito = 1;

    OPEN cursore;

    WHILE finito = 0 DO
        FETCH cursore INTO time, sensore;

        UPDATE statosensore SS
        SET Stato = "Effettuato"
        WHERE SS.Time = time
            and SS.sensore = sensore;

    END WHILE;
END $$

```

12.25 Creazione account dopo la registrazione

```

DROP TRIGGER IF EXISTS trigger_Account;
DELIMITER $$
CREATE TRIGGER trigger_Account AFTER INSERT ON registrato
FOR EACH ROW
BEGIN
    DECLARE password INTEGER;

    SELECT ROUND(RAND() * 1000)
    INTO password;

    INSERT INTO account
    VALUES (CONCAT(NEW.nome,NEW.cognome), password, current_date, null, null);
END $$

```

12.26 Gestione degli acquisti

Tramite il seguente trigger si ha la possibilità di gestire gli acquisti degli utenti. Se l'utente richiede un prodotto la cui quantità non è disponibile, l'ordine viene posto nello stato "pendente", in caso contrario sarà in "processazione". (Vincolo aggiunto: ogni account può avere un solo ordine nello stato pendente)

```
DROP TRIGGER IF EXISTS trigger_Acquisti;
DELIMITER $$
CREATE TRIGGER trigger_Acquisti BEFORE INSERT ON selezione
FOR EACH ROW
BEGIN
    DECLARE quantiDisponibili INTEGER DEFAULT 0;
    DECLARE stato VARCHAR(50) DEFAULT "";
    DECLARE quantita INT DEFAULT 0;
    DECLARE acquistato INT DEFAULT 0;
    DECLARE codAccount INT DEFAULT 0;
    DECLARE countPendeze INT DEFAULT 0;

    SET quantita = NEW.Quantita;

    -- conta quante unità sono disponibili
    SELECT COUNT(*) INTO quantiDisponibili
    FROM UnitaProdotto
    WHERE ProdottoCaseario = NEW.ProdottoCaseario
        AND Ordine IS NULL;

    -- seleziona l'account che ha effettuato l'ordine
    SELECT O.Account
    FROM Ordine O
    WHERE O.Codice = NEW.Ordine
    INTO codAccount;

    -- se la quantità voluta è maggiore di quella disponibile,
    -- lo stato sarà in 'pendente'
    IF quantita > quantiDisponibili THEN
        SELECT COUNT(*) INTO countPendenze
        FROM Ordine O
        WHERE O.Account = codAccount
            AND
            O.stato = "Pendente";

        IF countPendenze = 0 THEN
            SET stato = "Pendente";
        ELSE
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = "Attenzione, non puoi avere più di un ordine
                                nello stato pendente!";
        END IF;
    END IF;
```

```

ELSE
    SET stato = "Processazione";

    WHILE quantita <> 0 DO

        -- Selezioniamo il prodotto da vendere
        SELECT Codice INTO Acquistato
        FROM UnitaProdotto
        WHERE ProdottoCaseario = NEW.ProdottoCaseario
            AND Ordine IS NULL
        LIMIT 1;

        -- immettiammo nel campo Ordine del prodotto venduto il codice
        -- dell'ordine
        UPDATE unitaprodotto
        SET Ordine = NEW.Ordine
        WHERE Codice = Acquistato;

        SET quantita = quantita - 1;
    END WHILE;

END IF;

UPDATE Ordine O
SET O.Stato = stato
WHERE O.Account = codAccount
    AND O.Stato = "Pendente";

END $$

```


Con il seguente event, eseguito ogni 4 ore, si aggiorna lo stato dell'ordine.

```
DROP EVENT IF EXISTS event_Acquisti;
DELIMITER $$
CREATE EVENT event_Acquisti
ON SCHEDULE EVERY 4 HOUR
STARTS '2020-01-13 12:00:00'
DO
BEGIN
    DECLARE finito INTEGER DEFAULT 0;
    DECLARE statoCorrente VARCHAR(50) DEFAULT "";
    DECLARE statoSuccessivo VARCHAR(50) DEFAULT "";
    DECLARE Ordine INTEGER DEFAULT 0;

    DECLARE cursoreOrdini CURSOR FOR
        SELECT O.Stato, O.Ordine
        FROM ordine O
        WHERE O.Stato <> "Evaso";

    DECLARE CONTINUE HANDLER FOR NOT FOUND
        SET finito = 1;

    OPEN cursoreOrdini;

    WHILE finito = 0 DO
        FETCH cursoreOrdini INTO statoCorrente, Ordine;

        CASE
            WHEN statoCorrente = "Processazione" THEN
                SET statoSuccessivo = "In Preparazione";
            WHEN statoCorrente = "In Preparazione" THEN
                SET statoSuccessivo = "Spedito";
            WHEN statoCorrente = "Spedito" THEN
                SET statoSuccessivo = "Evaso";
        END CASE;

        UPDATE Ordine O
        SET Stato = statoSuccessivo
        WHERE O.Codice = Ordine;

    END WHILE;
    CLOSE cursoreOrdini;
END $$
```

12.27 Gestione inserimento dell'unità prodotto negli scaffali

Questo trigger permette di gestire l'inserimento delle unità prodotto negli scaffali. La prima cosa che occorre fare è quella di controllare se il prodotto deve essere posto in cantina o in magazzino. Una volta risolto questo, si ricerca lo scaffale che rispetta i vincoli richiesti come da specifica e si inseriscono i prodotti.

```
-- trigger inserimento unita prodotti negli scaffali
DROP TRIGGER IF EXISTS InserimentoScaffali_Cantina_Magazzino;
DELIMITER $$
CREATE TRIGGER InserimentoScaffali_Cantina_Magazzino
BEFORE INSERT ON unitaprodotto
FOR EACH ROW
BEGIN
    DECLARE isStagionatura VARCHAR(1) DEFAULT '';
    DECLARE scaffale INT DEFAULT 0;

    SET isStagionatura = ( -- se vale M --> Magazzino, C --> Cantina
        SELECT IF(R.DataStagionatura IS NULL, 'M', 'C')
        FROM ricetta R
        WHERE R.Prodotto = NEW.ProdottoCaseario
    );

    IF isStagionatura = 'M' THEN
        SET scaffale = (
            SELECT S.Codice
            FROM unitaprodotto U
            RIGHT OUTER JOIN
            scaffalemagazzino S ON U.ScaffaleMagazzino = S.Codice
            WHERE ( -- se il lotto è uguale o non ci sono prodotti
                U.Lotto = NEW.Lotto
            OR
            (
                SELECT COUNT(*)
                FROM scaffalemagazzino S2
                LEFT OUTER JOIN
                unitaprodotto U2 ON U2.ScaffaleMagazzino = S2.Codice
                WHERE S2.Codice = S.Codice
                AND U2.Lotto <> NEW.Lotto
            ) = 0
            )
            AND -
            -- e se il numero prodotti nello scaffale è < di CapienzaMax
            (
                SELECT IF(COUNT(*) < S3.CapienzaMax,1,0)
                FROM scaffalemagazzino S3
                LEFT OUTER JOIN
                unitaprodotto U3 ON U3.ScaffaleMagazzino = S3.Codice
                WHERE S3.Codice = S.Codice
```

```

        )
        LIMIT 1
    );
    SET NEW.ScaffaleMagazzino = scaffale;

ELSE -- altrimenti sarà da inserire nella cantina
    SET scaffale = (
        SELECT S.Codice
        FROM    unitaprodotto U
               RIGHT OUTER JOIN
               ScaffaleCantina S ON U.ScaffaleCantina = S.Codice
        WHERE   ( -- se il lotto è uguale o non ci sono prodotti
                  U.Lotto = NEW.Lotto
                OR
                (
                    SELECT COUNT(*)
                    FROM    ScaffaleCantina S2
                           LEFT OUTER JOIN
                           unitaprodotto U2 ON U2.ScaffaleCantina = S2.Codice
                    WHERE   S2.Codice = S.Codice
                           AND U2.Lotto <> NEW.Lotto
                ) = 0
                )
        AND -
        - e se il numero prodotti nello scaffale è < di CapienzaMax
        (
            SELECT IF(COUNT(*) < S3.CapienzaMax,1,0)
            FROM    ScaffaleCantina S3
                   LEFT OUTER JOIN
                   unitaprodotto U3 ON U3.ScaffaleCantina = S3.Codice
            WHERE   S3.Codice = S.Codice
        )
        LIMIT 1
    );

    SET NEW.ScaffaleCantina = scaffale;
    -- settiamo la datainizio della stagionatura
    SET NEW.DataInizio = current_date();

END IF;

END $$

```

13 Analytics

1) Comportamento degli animali

La seguente analytics permette di analizzare le posizioni di ogni singolo animale durante il pascolo. Infatti, preso in ingresso un pascolo, esamina tutte le posizioni degli animali ed inserisce una mangiatoia nella parte di zona dove gli animali preferiscono pascolare (come descritto nelle specifiche)

```
DROP PROCEDURE IF EXISTS analisi_pascolo;
DELIMITER $$
-- passiamo come parametro il prodotto
CREATE PROCEDURE analisi_pascolo (IN _codicePascolo VARCHAR(50))
BEGIN

    DECLARE timeInizio_pascolo_target TIMESTAMP DEFAULT NULL;
    DECLARE timeFine_pascolo_target TIMESTAMP DEFAULT NULL;
    DECLARE locale_target INT DEFAULT 0;

    DECLARE mungitrice_target INT DEFAULT 0;
    DECLARE long_mungitrice FLOAT DEFAULT 0;
    DECLARE lat_mungitrice FLOAT DEFAULT 0;

    -- inseriamo nelle rispettive variabili i dati del pascolo fornito in input
    SELECT P.TimeInizio, P.TimeFine, Locale
    FROM pascolo P
    WHERE P.Codice = _codicePascolo
    INTO timeInizio_pascolo_target, timeFine_pascolo_target, locale_target;

    /* raggruppiamo per l'intero (arrotondato) di longitudine e latitudine
       e ne contiamo il numero di ognuno;
       dopo di che prendiamo quelle più numerose
    */

    -- temp. tab. per le longitudini
    DROP TEMPORARY TABLE IF EXISTS _gpsTarget_long;
    CREATE TEMPORARY TABLE _gpsTarget_long (
        Longitudine INT NOT NULL PRIMARY KEY,
        Numero INT
    );

    INSERT INTO _gpsTarget_long
    SELECT ROUND(G.Longitudine),
           COUNT(*)
    FROM gps G
    INNER JOIN
        animale A ON G.Animale = A.Codice
    WHERE G.Time BETWEEN timeInizio_pascolo_target
                                   AND timeFine_pascolo_target
        AND
```

```

        A.CodiceLocale = locale_target
    GROUP BY ROUND(G.Longitudine);

    -- temp. tab. per le latitudini
    DROP TEMPORARY TABLE IF EXISTS _gpsTarget_lat;
    CREATE TEMPORARY TABLE _gpsTarget_lat (
        Latitudine INT NOT NULL PRIMARY KEY,
        Numero INT
    );

    INSERT INTO _gpsTarget_lat
    SELECT ROUND(G.Latitudine),
           COUNT(*)
    FROM gps G
    INNER JOIN
        animale A ON G.Animale = A.Codice
    WHERE G.Time BETWEEN timeInizio_pascolo_target
           AND timeFine_pascolo_target
           AND
           A.CodiceLocale = locale_target
    GROUP BY ROUND(G.Latitudine);

    -- prendiamo come longitudine quella più 'numerosa'
    SET long_mungitrice = (
        SELECT L.Longitudine
        FROM _gpsTarget_long L
        WHERE L.Numero >= ALL (
            SELECT L2.Numero
            FROM _gpsTarget_long L2
        )
        LIMIT 1
    );

    -- prendiamo come latitudine quella più 'numerosa'
    SET lat_mungitrice = (
        SELECT L.Latitudine
        FROM _gpsTarget_lat L
        WHERE L.Numero >= ALL (
            SELECT L2.Numero
            FROM _gpsTarget_lat L2
        )
        LIMIT 1
    );

    -
    - prendiamo come mungitrice quella più recentemente immessa nella tabella
    SET mungitrice_target = (
        SELECT G.Mungitrice
        FROM gps G

```

```

WHERE    G.Time >= ALL ( -- andiamo a vedere l'ultima sua posizione
SELECT   G2.Time
FROM     gps G2
WHERE    G2.Mungitrice = G.Mungitrice
)
LIMIT 1
);

INSERT INTO gps(Longitudine, Latitudine, Time, Mungitrice)
VALUES(long_mungitrice, lat_mungitrice, current_timestamp(),
                                             mungitrice_target);

END $$

```

2) Tracciabilità di filiera

La seguente analytics permette di ripercorrere a ritroso la filiera alla ricerca di eventuali problematiche rilevate nel prodotto, deducibili da resi. Se il numero di resi è maggiore o uguale al 60% delle unità ordinate, allora vengono eliminati tutti i prodotti unitari di quel prodotto caseario e vengono modificati i parametri ideali delle fasi del prodotto tenendo conto dei parametri di qualità del prodotto analizzati dopo il reso (ultimo reso effettuato). Si tiene conto del fatto che i parametri ideali delle fasi ricetta sono gli stessi dei parametri di qualità dei resi.

```
DROP PROCEDURE IF EXISTS analisi_tracciabilita;
DELIMITER $$
-- passiamo come parametro il prodotto
CREATE PROCEDURE analisi_tracciabilita (IN _prodotto VARCHAR(50))
BEGIN

    DECLARE countRecensioniNeg INT DEFAULT 0;
    DECLARE countResi INT DEFAULT 0;
    DECLARE dataUltimoReso DATE DEFAULT NULL;
    DECLARE venditeCurrentMonth INT DEFAULT 0;

    -- calcoliamo il numero di recensioni negative del prodotto del mese in corso
    SELECT COUNT(*)
    FROM recensione R
        INNER JOIN
            unitaprodotto U ON R.UnitaProdotto = U.Codice
        INNER JOIN
            prodottocaseario P ON U.ProdottoCaseario = P.Nome
    WHERE P.Nome = _prodotto
        AND
        (
            R.Gradimento < 5
            OR R.Qualita < 5
            OR R.Gusto < 5
        )
        AND
        (
            YEAR(CURRENT_DATE) = YEAR(R.Data)
            AND MONTH(R.Data) = MONTH(CURRENT_DATE)
        )
    INTO countRecensioniNeg;

    -
    -- calcoliamo il numero di resi del prodotto e la data dell'ultimo reso (mese corrente)
    SELECT COUNT(*), MAX(R.DataReso)
    FROM reso R
        INNER JOIN
            unitaprodotto U ON R.UnitaProdotto = U.Codice
        INNER JOIN
            prodottocaseario P ON U.ProdottoCaseario = P.Nome
```

```

WHERE    P.Nome = _prodotto
        AND
        (
            YEAR(CURRENT_DATE) = YEAR(R.DataReso)
            AND MONTH(R.DataReso) = MONTH(CURRENT_DATE)
        )
INTO countResi, dataUltimoReso;

-- se il numero dei resi è maggiore o uguale del 60% dei prodotti ordinati
-- allora eliminiamo tutti i prodotti unitari di quel prodotto

-- numero vendite prodotto mese attuale
SELECT SUM(S.Quantita)
FROM    selezione S
        INNER JOIN
        ordine O ON S.Ordine = O.Codice
WHERE    (
            YEAR(CURRENT_DATE) = YEAR(O.Time)
            AND MONTH(O.Time) = MONTH(CURRENT_DATE)
        )
        AND S.ProdottoCaseario = _prodotto
INTO venditeCurrentMonth;

-- cancellazione se rispetta la condizione
IF (countResi >= 0.6*venditeCurrentMonth) AND (countRecensioniNeg > 0) THEN
    DELETE U.*
    FROM    unitaprodotto U
    WHERE    U.ProdottoCaseario = _prodotto
            AND U.Ordine IS NULL;

/*
Adesso andiamo a modificare i parametri ideali delle fasi del prodotto
tenendo conto dei parametri di qualità del prodotto analizzati dopo
il reso.

Creazione di una temporary table per i parametri del prodotto da
modificare
*/
DROP TEMPORARY TABLE IF EXISTS _parametro;
CREATE TEMPORARY TABLE _parametro(
    Parametro VARCHAR(50) NOT NULL PRIMARY KEY
);

INSERT INTO _parametro
    SELECT IF(PR.Valutazione < 5, PID.Parametro, '0')
    FROM
        ( -- parametri ideali e quantità
        SELECT C.Parametro, C.Quantita
        FROM    prodottocaseario P
                INNER JOIN

```



```

        composta C ON C.Prodotto = P.Nome
WHERE    P.Nome = _prodotto
) AS PID -- par. ideali
NATURAL JOIN
( -
- parametri qualita e valutazione sull'ultimo reso effettuato
    SELECT C.ParametroQualita AS Parametro, C.Valutazione
    FROM    controllo C
            INNER JOIN
            reso R ON C.Reso = R.Codice
            INNER JOIN
            unitaprodotto U ON R.UnitaProdotto = U.Codice
            INNER JOIN
            prodottocaseario P ON U.ProdottoCaseario = P.Nome
    WHERE   P.Nome = _prodotto
            AND R.DataReso = dataUltimoReso
    ) AS PR -- par. reso
GROUP BY PID.Parametro;

-- Variazione della quantita del parametro (da -10 a + 10)
UPDATE  composta C
        NATURAL JOIN
        _parametro PP
SET      C.Quantita = C.Quantita + (ROUND(RAND()*(-10) + RAND()*(10)))
WHERE    C.Prodotto = _prodotto;

END IF;

END $$

```

3) Analisi vendite di un prodotto caseario

La seguente analytics permette di analizzare le vendite di un prodotto caseario. Riceve come input il prodotto e successivamente ne calcola le vendite sia del mese corrente che del mese precedente e le confronta. Se le vendite del mese corrente sono incrementate del 30% o più rispetto a quelle del mese precedente e non ci sono stati eventuali resi, allora il costo del prodotto viene aumentato del 5%.

Invece, se le vendite del mese corrente sono diminuite del 50% o più rispetto a quelle del mese precedente, allora il costo del prodotto viene diminuito del 10%.

```
DROP PROCEDURE IF EXISTS analisi_vendite;
DELIMITER $$
-- passiamo come parametro il prodotto
CREATE PROCEDURE analisi_vendite (IN _prodotto VARCHAR(50))
BEGIN

    DECLARE venditeCurrentMonth INT DEFAULT 0;
    DECLARE venditePastMonth INT DEFAULT 0;
    DECLARE countResi INT DEFAULT 0;

    -- numero vendite prodotto mese attuale
    SELECT SUM(S.Quantita)
    FROM selezione S
    INNER JOIN
        ordine O ON S.Ordine = O.Codice
    WHERE (
        YEAR(CURRENT_DATE) = YEAR(O.Time)
        AND MONTH(O.Time) = MONTH(CURRENT_DATE)
    )
        AND S.ProdottoCaseario = _prodotto
    INTO venditeCurrentMonth;

    -
    - calcoliamo il numero di resi del prodotto e la data dell'ultimo reso (mese corrente)
    SELECT COUNT(*)
    FROM reso R
    INNER JOIN
        unitaprodotto U ON R.UnitaProdotto = U.Codice
    INNER JOIN
        prodottocaseario P ON U.ProdottoCaseario = P.Nome
    WHERE P.Nome = _prodotto
        AND
        (
            YEAR(CURRENT_DATE) = YEAR(R.DataReso)
            AND MONTH(R.DataReso) = MONTH(CURRENT_DATE)
        )
    INTO countResi;
```

-
- aumento costo prodotto se le vendite del mese corrente sono incrementate del 30% rispetto a quelle del mese precedente

```
SELECT SUM(S.Quantita)
FROM   selezione S
      INNER JOIN
ordine O ON S.Ordine = O.Codice
WHERE  (
      IF(MONTH(CURRENT_DATE) = 1,
        YEAR(CURRENT_DATE) - 1 = YEAR(O.Time)
        AND MONTH(O.Time) = 12,
        YEAR(CURRENT_DATE) = YEAR(O.Time)
        AND MONTH(O.Time) = MONTH(CURRENT_DATE)-1)
      )
      AND S.ProdottoCaseario = _prodotto
INTO venditePastMonth;

-- aumento del costo del prodotto del 5%
IF (venditeCurrentMonth > venditePastMonth*1.3) AND countResi = 0 THEN
  UPDATE prodottoCaseario
  SET costo = costo * 1.05 -- aumento del 5%
  WHERE Nome = _prodotto;
-- diminuzione del costo del prodotto del 10%
ELSEIF (venditeCurrentMonth <= venditePastMonth*0.5) AND countResi > 0 THEN
  UPDATE prodottoCaseario
  SET costo = costo - costo * 0.1 -- diminuzione del 10%
  WHERE Nome = _prodotto;
END IF;

END $$
```