



Università degli Studi di Salerno



Dipartimento di Ingegneria dell'Informazione ed Elettrica e
Matematica Applicata

Corso di Laurea in Ingegneria Informatica

Basi di Dati 2023/2024 Canale A-H

Project Work

Traccia A13- GAMTech

Gruppo n. **12 – Canale AH**

WP	Cognome e Nome	Matricola	e-mail	Responsabile
1	Allocca Francesco Vittorio	0612706382	f.allocca5@studenti.unisa.it	X
2	Foschillo Gennaro	0612705941	g.foschillo@studenti.unisa.it	

Anno accademico 2023-2024

Sommario

1. Descrizione della realtà di interesse	3
1.1. Analisi della realtà di interesse	3
2. Analisi delle specifiche	5
2.1. Glossario dei termini	5
2.2. Strutturazione dei requisiti in frasi	5
2.2.1. Frasi di carattere generale	5
2.2.2. Frasi relative a GAMBERAIA	5
2.2.3. Frasi relative a TRATTAMENTO	5
2.2.4. FRASI RELATIVE A TIPI SPECIFICI DI SENSORE	6
2.2.5. FRASI RELATIVE A TIPI SPECIFICI DI PARAMETRO	6
2.2.6. FRASI RELATIVE A TIPI SPECIFICI DI ALLARME	6
2.2.7. FRASI RELATIVE A TIPI SPECIFICI DI GAMBERO	6
2.2.8. FRASI RELATIVE A TIPI SPECIFICI DI GIORNATA DI PRELIEVO	6
2.3. Identificazione delle operazioni principali	7
3. Progettazione Concettuale	8
3.1. Schema Concettuale	8
3.2. Design Pattern	8
3.2.1. Pattern DI REIFICAZIONE BINARIA (TRATTAMENTO-APPLICAZIONE- GAMBERAIA)	8
3.2.2. Pattern DI REIFICAZIONE BINARIA (GAMBERAIA -MISURAZIONE- PARAMETRO)	9
3.2.3. Pattern DI REIFICAZIONE DI ATTRIBUTO DI ENTITA' (SENSORE -AZIENDA)	10
3.2.4. Pattern DI REIFICAZIONE DI STORICIZZAZIONE DI RELAZIONE (GAMBERAIA -OSPITALITA'-GAMBERO)	11
3.3. Dizionario dei Dati	11
3.4. Regole Aziendali	17
4. Progettazione Logica	19
4.1. Ristrutturazione Schema Concettuale	19
4.1.1. Analisi delle Prestazioni	19
4.2. Analisi delle ridondanze	21
4.2.1. Analisi della ridondanza 1: Numero Sensori	21
4.2.2. Analisi della ridondanza 2: Quantità Totale in Giornata di prelievo	22
4.3. Eliminazione delle generalizzazioni	24
4.3.1. Generalizzazione <i>Sensore</i>	24
4.3.2. Generalizzazione <i>Ospitalità</i>	25
4.4. Partizionamento/Accorpamento Entità e Associazioni	26
4.4.1. Eliminazione attributi multi-valore e composti	26
4.4.2. Accorpamento Entità / Associazioni	27
4.5. Schema ristrutturato finale	29
.....	29

4.6.	Schema logico.....	30
4.7.	Documentazione dello schema logico.....	32
5.	Normalizzazione	33
	Script Creazione e Popolamento Database	40
5.1.	Query con operatore di aggregazione e join: Media della quantità di gamberi prelevati per ogni specie e numero di prelievi effettuati	50
5.2.	Query nidificata complessa: Trova tutti i trattamenti applicati a una specifica gamberaia in un anno, con i relativi materiali utilizzati e le quantità	51
5.3.	Query insiemistica: Sensori applicati ad una particolare Gamberaia	51
6.	Viste	52
6.1.	Vista Vista Gamberaie, Gamberi e Trattamenti Associati.....	52
6.1.1.	Query con Vista: Elenco delle gamberaie con relativi gamberi e trattamenti	52
7.	Trigger.....	52
7.1.	Trigger inizializzazione.....	52
7.1.1.	Trigger Generazione codice univoco per Gamberaia	52
7.1.2.	Trigger per valori di default sensore	53
7.2.	Trigger per vincoli aziendali.....	53
7.2.1.	Trigger1: Aggiornamento dell'ospitalità	54
7.2.2.	Trigger per misurazione non valida	54
7.2.3.	Trigger aggiornamento quantità totale	55

1. Descrizione della realtà di interesse

Titolo: **GAMTech**

Nel contesto dell'acquacoltura dei gamberi, progettiamo un sistema di gestione che tenga traccia delle informazioni essenziali per la coltivazione di questi crostacei. Di seguito sono riportate le specifiche chiave:

Ogni gamberaia è un'area specifica in cui vengono coltivati i gamberi. Ogni gamberaia è identificata da un codice e può avere una posizione geografica specifica. Durante una particolare stagione, una gamberaia può ospitare una sola tipologia di gamberi. Alcune gamberaie potrebbero essere momentaneamente non coltivate. Ogni tipo di gambero coltivato è identificato un nome comune, un nome scientifico, una varietà, una categoria, ecc. Ad esempio, potremmo coltivare Gamberi Rossi (nome comune) con il nome scientifico *Penaeus monodon* (nome scientifico) nella categoria "Gamberi Giganti".

I gamberi richiedono trattamenti specifici per la loro crescita e salute. I trattamenti necessari per la coltivazione dei gamberi sono identificati da un codice univoco e un nome dell'attività. Per ogni trattamento, si tiene traccia dei materiali/prodotti necessari e delle relative quantità. I trattamenti vengono effettuati su ogni gamberaia, con registrazione della data, dell'operatore e della gamberaia interessata. Diverse gamberaie possono ricevere gli stessi trattamenti, e un trattamento può essere effettuato su più gamberaie.

Per monitorare i parametri vitali dei gamberi allevati, si utilizzano sensori specifici per l'acquacoltura dei gamberi. Ogni sensore è caratterizzato da una marca, un modello, l'azienda produttrice, una descrizione e il parametro che rileva (ad esempio, temperatura dell'acqua, salinità, ossigeno disciolto, ecc.). Ogni gamberaia può avere più sensori, anche dello stesso tipo. Ogni sensore ha un codice, un indirizzo di rete, una posizione all'interno della gamberaia e un consumo di energia medio. Le misurazioni effettuate dai sensori sono registrate insieme alla data orario, valore, unità di misura.

Esistono dei valori soglia per alcuni parametri. Al superamento di un valore soglia, deve essere memorizzato un allarme. Tali allarmi consentiranno agli addetti di decidere i trattamenti da attuare. I gamberi coltivati sono prelevati in una particolare giornata di confezionamento. Per una giornata di prelievo si memorizza la quantità di gamberi prelevati da ogni vasca (così da poter valutare la produttività e l'adeguatezza dei trattamenti).

1.1. Analisi della realtà di interesse

L'obiettivo del progetto è quello di realizzare un database che permetta la gestione di un insieme di gamberaie in cui avviene l'allevamento di determinate specie di gambero. Per ogni gamberaia, si tiene traccia di un codice univoco che la identifica, il tipo di gambero allevato, la stagione, capienza massima e una posizione geografica specifica, che deve essere diversa per ogni gamberaia.

L'allevamento dei gamberi è vincolato dalla stagione di allevamento associata ad ogni specie. Le gamberaie possono consentire l'allevamento di massimo una tipologia di gambero durante una stagione, e più gamberaie possono ospitare la stessa specie contemporaneamente. Le specie di gambero attualmente in allevamento vengono monitorate da determinati sensori che hanno il compito di misurare i parametri relativi alle specie allevate e al loro ambiente di allevamento. I sensori vengono descritti da una marca, un modello e identificati da un codice univoco e una descrizione del parametro che essi misurano. Alcuni sensori possono misurare più parametri. In particolare, quando un sensore viene installato su una gamberaia viene salvato un indirizzo di rete, un consumo di energia medio e una posizione specifica all'interno della gamberaia.

I gamberi necessitano di trattamenti per la loro salute e per le necessità di allevamento. Nella gestione dell'allevamento è necessario effettuare trattamenti specifici su ogni gamberaia. Questi trattamenti sono gestiti da un operatore in un noto orario. Per ogni trattamento, si tiene traccia di un codice associato, una descrizione del trattamento e informazioni relative ai materiali utilizzati. Una gamberaia può ricevere più trattamenti anche dello stesso tipo ma ad orari diversi. I trattamenti utilizzati, possono essere determinati da particolari allarmi, che sono attivati al superamento di prestabiliti valori soglia associati ad ogni parametro. Inoltre, si tiene traccia di particolari giornate di prelievo, durante le quali viene analizzata la salute dei gamberi coltivati e si procede al confezionamento, registrando, anche la quantità di gamberi confezionati per poter così valutare l'adeguatezza e la produttività dei trattamenti.

2. Analisi delle specifiche

Workpackage	Task	Responsabile
WP0	Analisi delle specifiche	Intero Gruppo

2.1. Glossario dei termini

	Termine	Descrizione	Sinonimi	Collegamenti
1	Gamberaia	L'area specifica in cui vengono allevati i gamberi	Vasca	Gambero , Sensore, Trattamento
2	Trattamento	Azioni specifiche effettuate su ogni gamberaia	Intervento	Operatore, Gamberaia
3	Sensore	Permette di rilevare determinati parametri all'interno della gamberaia	-	Parametro, Gamberaia, Azienda
4	Parametro	Tipo di misura effettuata da uno specifico sensore	Misura	Sensore, Misurazione
5	Allarme	Segnale che viene generato quando si verificano determinate condizioni relative alle misurazioni. Richiede l'intervento degli operatori.	Segnale, Avviso	Operatore, Misurazioni
6	Gambero	Rappresenta una specie di crostaceo allevata all'interno delle gamberaie	Crostaceo	Gamberaia
7	Giornata di prelievo	Data precisa in cui viene effettuato il prelievo dei gamberi per un controllo e per avviare il confezionamento di essi	Giornata di controllo	Gamberaia

Tabella 1. Glossario dei Termini

2.2. Strutturazione dei requisiti in frasi

2.2.1. Frasi di carattere generale

Nel contesto dell'acquacoltura dei gamberi, si vuole progettare un sistema di gestione che tenga traccia delle informazioni essenziali per la coltivazione e monitoraggio di questi crostacei.

2.2.2. Frasi relative a GAMBERAIA

Per definire una gamberaia dobbiamo definire un codice univoco, una capienza massima, Per la posizione della gamberaia si va a definire una longitudine e una latitudine. Durante una particolare stagione, una gamberaia può ospitare una sola tipologia di gamberi. I trattamenti e l'allevamento dei crostacei sono svolti presso le gamberaie.

2.2.3. Frasi relative a TRATTAMENTO

Relativamente ai trattamenti si va a descriverli con un codice univoco che identifica il tipo di trattamento, un nome dell'attività che va a descrivere l'attività effettuata.

Inoltre, si tiene traccia dei prodotti per ogni trattamento descritti dai materiali utilizzati e dalle singole quantità di ciascun materiale utilizzati. Si noti che ad ogni trattamento possono essere

associati più prodotti, e ad ogni prodotto è indicato una descrizione del materiale utilizzato e la quantità di esso.

2.2.4. FRASI RELATIVE A TIPI SPECIFICI DI SENSORE

Per ogni sensore si caratterizza una marca, associata al nome dell'azienda produttrice, un modello, una descrizione funzionale e un codice identificativo univoco. Quando viene installato in una gamberaia, il sensore è associato a uno specifico indirizzo di rete e a un consumo medio di energia. Inoltre, il sensore oltre che con l'indirizzo di rete, si identifica con il codice della gamberaia dove esso è stato installato, e la posizione all'interno di essa (in alto, in basso, al centro, etc.)

2.2.5. FRASI RELATIVE A TIPI SPECIFICI DI PARAMETRO

Ogni sensore può misurare un solo parametro, o essere multi-parametrico, ovvero in grado di rilevare più parametri. Ogni parametro è obbligatoriamente associato a un sensore e viene rappresentato da una descrizione che ne specifica la natura della misurazione, un valore massimo soglia oltre il quale si considera un'anomalia e un'unità di misura associata al parametro stesso.

2.2.6. FRASI RELATIVE A TIPI SPECIFICI DI ALLARME

Quando viene superato un valore soglia di una misurazione associato a un parametro, viene attivato un allarme. Questo allarme tiene traccia della misurazione che ha scatenato l'allarme, del parametro a cui è associata la misurazione e di un ID univoco associato all'allarme stesso.

2.2.7. FRASI RELATIVE A TIPI SPECIFICI DI GAMBERO

Ogni gambero è rappresentato nel suo complesso come una tipologia di gambero. Pertanto, in una specifica gamberaia può esserci solo un'occasione di tipo di gambero per stagione di allevamento. Un gambero è caratterizzato da una categoria, che rappresenta l'attributo più generale, un nome comune e un nome scientifico che identificano la tipologia. Inoltre, è presente l'attributo varietà, che indica la zona dove quel gambero è comune, e l'attributo stagione di allevamento, che rappresenta la stagione in cui la tipologia di gambero è coltivabile. Un esempio potrebbe essere: Categoria: Gamberi Giganti, Nome comune: Gambero Rosso, Nome scientifico: *Aristaeomorpha foliacea*, Varietà: Gamberi Rossi del Mediterraneo, Stagione di allevamento: Estate

2.2.8. FRASI RELATIVE A TIPI SPECIFICI DI GIORNATA DI PRELIEVO

In una specifica giornata di prelievo si determina la qualità dei gamberi tenendo traccia della data in cui vengono prelevati, della loro idoneità e del codice della gamberaia in cui essi vengono prelevati. Si registra la quantità di gamberi prelevati da ogni vasca per valutare la produttività e l'efficacia dei trattamenti.

2.3. Identificazione delle operazioni principali

Operazione 1: Assegnare una tipologia di Gambero in una gamberaia libera (in media 40 volte a stagione)

Operazione 2: Assegnamento di un trattamento indicando la gamberaia su cui è associato con relativi dati: inserimento nome dell'attività, codice, materiali utilizzati e la relativa quantità (operazione svolta mediamente 500 volte al giorno)

Operazione 3: trova i dati di una gamberaia, la tipologia gambero che attualmente alleva e i trattamenti che sono stati applicati ad essi in una specifica stagione

Operazione 4: Trova tutte le gamberaie che coltivano in quella stagione una particolare tipologia di gambero (1 volte a stagione)

Operazione 5: inserisci i dati di un nuovo sensore applicato ad una specifica gamberaia, specificando l'indirizzo di rete e la posizione in cui è applicata ed il consumo di energia e tutti gli attributi di sensore (operazione effettuata 750 volte a stagione)

Operazione 6: Salvataggio di una misurazione associato a un parametro di un sensore applicato su una gamberaia (operazione effettuata 24 volta al giorno)

Operazione 7: Salvataggio di un nuovo tipo di allarme rilevato e dei suoi dati, inclusa la relativa misurazione (operazione da effettuare circa 150 volte al giorno)

Operazione 8: Assegna un operatore per gestire un allarme rilevato e definisci un trattamento relativo (operazione da effettuare circa 150 volte al giorno)

Operazione 9: Stampa tutti i dati relativi a una giornata di prelievo, inclusa la quantità totale di gamberi totali prelevati in essa (1 volta al mese)

****consideriamo il caso in cui in una stagione vengano popolate un massimo di 50 gamberaie**

3. Progettazione Concettuale

Workpackage	Task	Responsabile
WP1	Progettazione Concettuale	Allocca Francesco Vittorio

3.1. Schema Concettuale

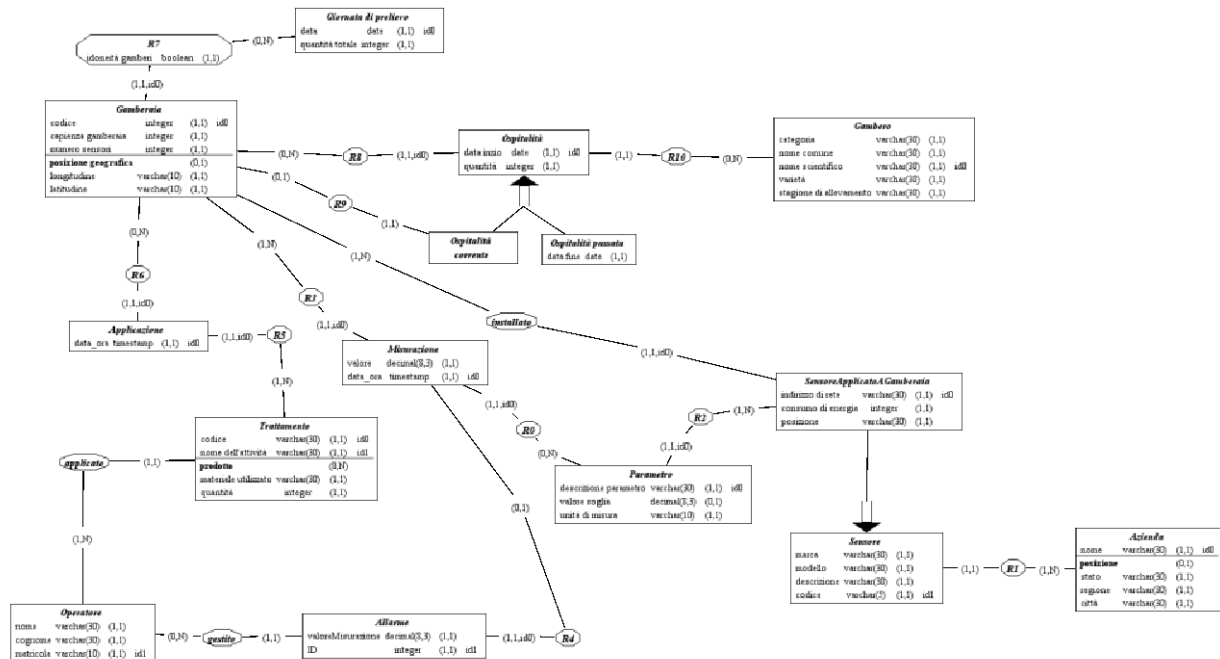


Figura 1 Schema E-R

3.2. Design Pattern

In seguito, sono stati descritti i pattern applicati nella progettazione concettuale.

3.2.1. Pattern DI REIFICAZIONE BINARIA (TRATTAMENTO-APPLICAZIONE-GAMBERAIA)



Figura 2 prima di applicare il pattern 3.2.1

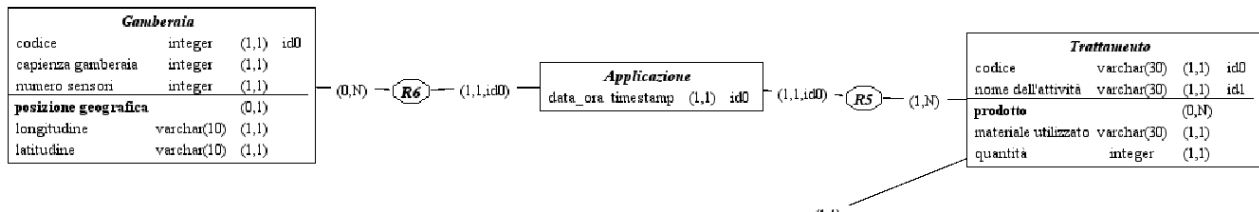


Figura 3 pattern applicato

In precedenza, era impossibile associare più volte lo stesso trattamento alla stessa gamberaia, poiché ciò avrebbe generato duplicati nei valori identificativi, risultando in occorrenze identiche. Tuttavia, con l'applicazione di questo pattern, tale vincolo è stato superato mediante l'attribuzione di identificatori agli attributi che connettono le entità. Pertanto, ogni trattamento è ora vincolato a una gamberaia con un orario differente (sia data che ora), consentendo l'applicazione ripetuta del medesimo trattamento alla medesima gamberaia con la condizione che almeno la data o l'ora siano diversi.

3.2.2. Pattern DI REIFICAZIONE BINARIA (GAMBERAIA -MISURAZIONE-PARAMETRO)

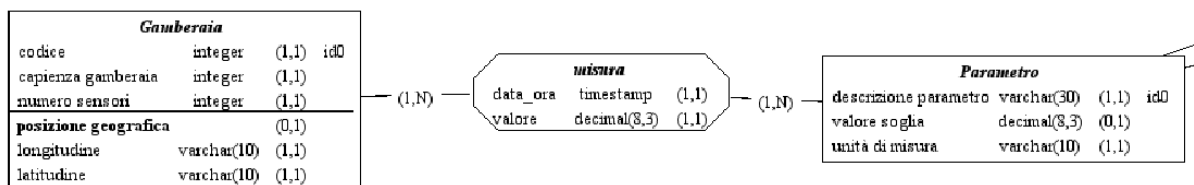


Figura 4 prima di applicare il pattern 3.2.2

Questo pattern viene impiegato affinché lo stesso parametro possa effettuare misurazione più volte sulla stessa gamberaia, tuttavia, la data di misurazione deve essere differente. In precedenza, era impossibile avere due occorrenze dello stesso parametro stesso identificativo (descrizione parametro) associato alla stessa gamberaia. Tuttavia, grazie all'implementazione di questo pattern, anche la data e l'ora sono diventati identificativi. Di conseguenza, possono esistere due occorrenze dello stesso parametro con lo stesso identificativo associati alla stessa gamberaia, ma la data o l'ora di misurazione devono essere necessariamente diverse.

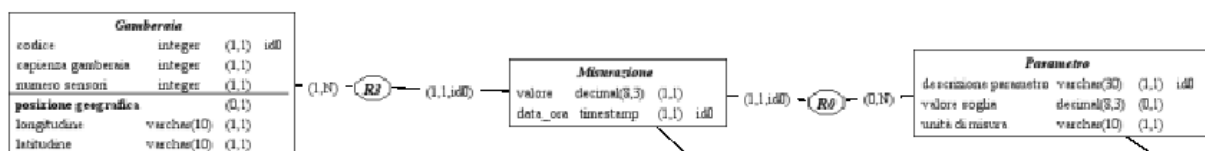


Figura 5 pattern applicato

3.2.3. Pattern DI REIFICAZIONE DI ATTRIBUTO DI ENTITA' (SENSORE -AZIENDA)

Approccio di design dove un concetto rappresentato da un attributo di un'entità viene elevato a un'entità separata. In altre parole, anziché rappresentare un concetto come un attributo di un'altra entità, viene modellato come un'entità distinta.

Questo pattern rende il modello dei dati più flessibile e riduce la ridondanza dei dati.

<i>Sensore</i>			
marca	varchar(30)	(1,1)	
modello	varchar(30)	(1,1)	
descrizione	varchar(30)	(1,1)	
codice	varchar(30)	(1,1)	id1
azienda	varchar(30)	(1,1)	

Figura 6 prima di applicare il pattern 3.2.3

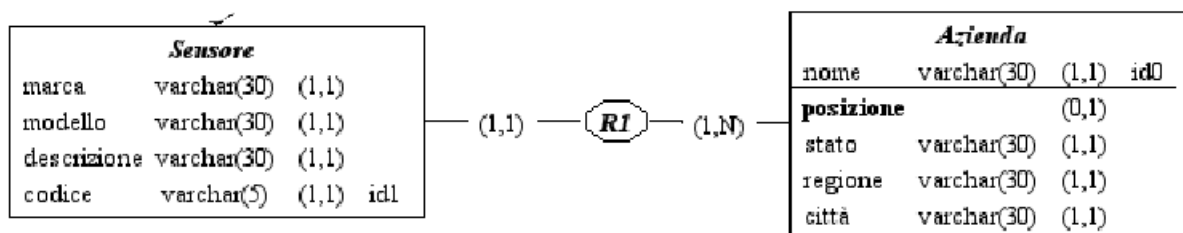


Figura 7 pattern applicato

3.2.4 Pattern DI REIFICAZIONE DI STORICIZZAZIONE DI RELAZIONE (GAMBERAIA -OSPITALITA'-GAMBERO)

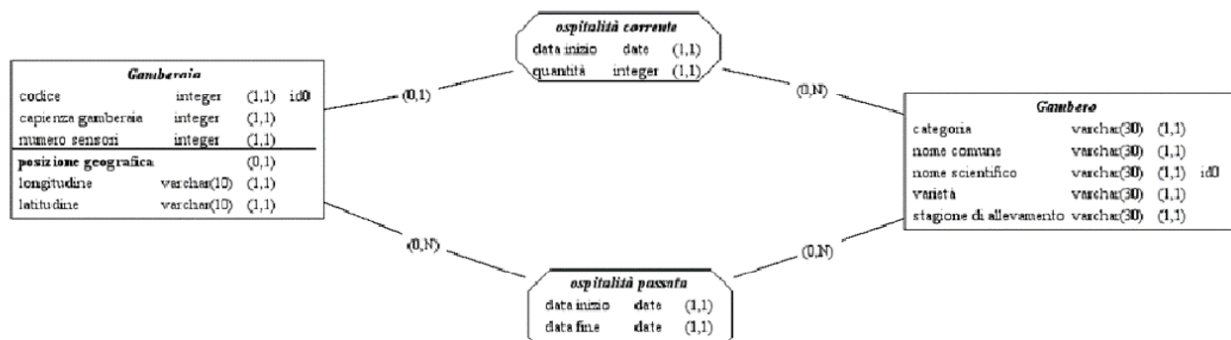


Figura 8 prima di applicare il pattern 3.2.4

Non è possibile rappresentare la situazione in cui lo stesso tipo di gambero viene ospitato in periodi diversi presso la stessa gamberaia, poiché ciò genererebbe occorrenze duplicate. Inoltre, non è possibile associare esclusivamente un tipo di gambero a una singola ospitalità. Per risolvere entrambe le problematiche, è necessario applicare il pattern di storicizzazione ed introdurre un vincolo esterno allo schema dei dati che assicuri che tutte le relazioni tra Gambero e Ospitalità corrente siano presenti anche tra le relazioni tra Gambero e Ospitalità passata.

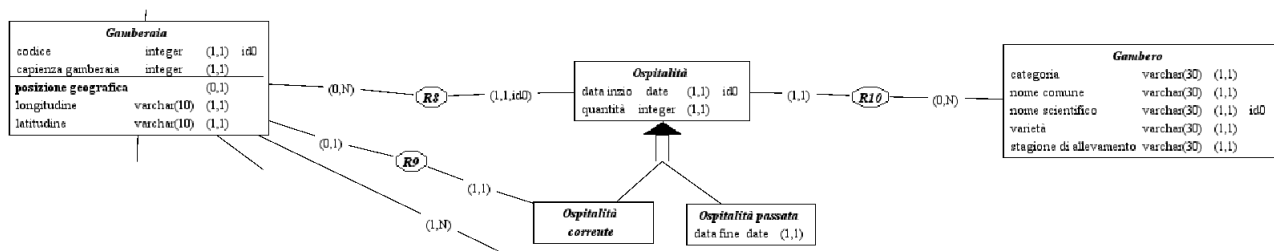


Figura 9 pattern applicato

3.3. Dizionario dei Dati

Entità	Descrizione	Attributi	Identificatore
Gamberaia	Area specifica in cui vengono allevati i crostacei	Codice Capienza gamberaia Numero sensori Posizione Longitudine Latitudine	-Codice
Gambero	La tipologia di gambero che viene	Categoria Nome Comune	-Nome scientifico

	coltivato nella gamberaia in una particolare stagione di allevamento	Nome Scientifico Varietà Stagione di allevamento	
Ospitalità	Attività che consiste nell'allevare una specifica tipologia di gambero, assegnando ad essa una gamberaia particolare.	Inizio Quantità	-Inizio
Ospitalità corrente	Attività attraverso la quale si fa riferimento a un allevamento attuale dei crostacei, fornendo una data di inizio e associandola a una specifica gamberaia, indicando così la sua attuale occupazione.		
Ospitalità passata	Attività relativa a un allevamento precedente di crostacei, caratterizzata da una data di inizio e una data di fine. Essa determina il periodo in cui è avvenuto l'allevamento dei gamberi in una particolare gamberaia.	Data fine	
Trattamento	Attività specifica volta a garantire la crescita e la salute dei gamberi allevati.	Codice Nome dell'attività Prodotti Materiale utilizzato Quantità	-Codice -Nome dell'attività
Applicazione	L'azione di compiere uno specifico trattamento a una specifica gamberaia	Data Ora	-Data, Ora
Operatore	Individuo responsabile dell'esecuzione dei	Nome Cognome Matricola	-Matricola

	trattamenti e della gestione degli allarmi		
Sensore	Dispositivo in grado di misurare uno o più parametri, non ancora applicato a una gamberaia, è associato ad una azienda dove è stato prodotto.	Marca Modello Descrizione Codice	-Codice
Sensore applicato a Gamberaia	Specializzazione di sensore quando viene installato all'interno di una gamberaia	Indirizzo di rete Consumo di energia Posizione all'interno della gamberaia	-Indirizzo di rete
Azienda	Un'impresa coinvolta nella vendita di diversi sensori	Nome Posizione Stato Regione Città	-Nome
Parametro	Caratteristica chimica o fisica che il sensore è in grado di rilevare	Descrizione parametro Valore soglia Unità di misura	- Descrizione parametro
Misurazione	Dato di misura derivato dal sensore durante l'acquisizione di una grandezza fisica specifica relativa a un particolare parametro	Data Ora Valore	-Data, Ora
Allarme	Avviso generato quando viene rilevato una misurazione che ha superato un valore soglia di un determinato parametro, richiede l'intervento di un operatore	Valore misurazione ID	-ID
Giornata di prelievo	Determinata giornata in cui si effettua il prelievo di gamberi per il confezionamento o per un relativo controllo	Quantità totale Data	-Data

Tabella 2. Dizionario dei dati – Entità

Relazioni	Descrizione	Entità Coinvolte	Attributi
R0	Una misurazione effettuata è associata ad un solo parametro. A quest'ultimo invece, possono essere associate più misurazioni.	Misurazione (1,1 , id0) Parametro (1,N)	
R1	Un particolare sensore esiste è associato ad un'azienda. Quest'ultima è associata a più tipi di sensore , e un sensore ad un'unica azienda identificata dal nome.	Azienda (1,N) Sensore (1,1)	
R2	Un determinato parametro è associato ad un unico sensore , quest'ultimo invece può rilevare uno o più parametri contemporaneamente.	Parametro (1,1 , id0) Sensore Applicato a Gamberaia (1,N)	
R3	Ad una gamberaia possono essere associate più misurazioni associate allo stesso parametro con orari diversi. Una particolare misurazione invece, è fatta su una specifica gamberaia identificata da un codice.	Gamberaia (1,N) Misurazione (1,1,id0)	
R4	Una misurazione può far scaturire un allarme in particolari condizioni, un'occorrenza di allarme deve essere associata	Misurazione (0,1) Allarme (1,1,id0)	

	necessariamente ad una misurazione effettuata.		
R5	Un'occorrenza di trattamento può essere associata a una applicazione una o più volte. L'applicazione di un trattamento può essere associata ad una sola occorrenza di quest'ultimo	Trattamento (1,N) Applicazione (1,1,id0)	
R6	L'applicazione di un trattamento in una determinata data e ora può essere associata a una singola gamberaia. Non possono esistere due occorrenze di applicazione dello stesso trattamento sulla medesima gamberaia con la stessa data e ora. Su una singola gamberaia, possono essere associate da 0 a N applicazioni di uno specifico trattamento.	Gamberaia (0,N) Applicazione (1,1,id0)	
R7	Una singola giornata di prelievo può essere associata a più gamberaie. Inoltre, ciascuna gamberaia è associata a una giornata di prelievo con una data specifica.	Gamberaia (1,1,id0) Giornata di prelievo (0,N)	-Idoneità gamberi
R8	Ad un'occorrenza di gamberaia possono essere associate da 0 a N ospitalità, nel caso in cui quest'ultima si specializzi come Ospitalità passata. Un'occorrenza di ospitalità deve essere associata	Ospitalità (1,1,id0) Gamberaia (0,N)	

	esclusivamente a una sola gamberaia.		
R9	Una gamberaia può attualmente ospitare nessuna o al massimo una ospitalità corrente. Allo stesso modo, ciascuna ospitalità corrente può essere associata esclusivamente a una sola gamberaia.	Gamberaia (0,1) Ospitalità corrente (1,1)	
R10	Un tipo di gambero può essere associato a più occorrenze di tipo Ospitalità. D'altra parte, ogni singola occorrenza di Ospitalità può associare un solo tipo di gambero.	Gambero (0,N) Ospitalità (1,1)	
Applicato	Un operatore può applicare uno o più occorrenze di trattamenti. Un trattamento deve essere applicato da un solo operatore	Trattamento (1,1) Operatore (1,N)	
Gestito	Un allarme deve essere gestito da uno operatore. Il singolo operatore può gestire da 0 a più allarmi	Operatore (1,N) Allarme (1,1)	
Installato	Un sensore applicato alla gamberaia è installato solo su una gamberaia specifica. Su una gamberaia possono essere installati più sensori.	Gamberaia (1,N) Sensore Applicato a Gamberaia (1,1,id0)	

Tabella 3. Dizionario dei dati - Relazioni

Workpackage	Task	Responsabile
WP4	Regole Aziendali	Foschillo Gennaro

3.4. Regole Aziendali

Regole di Vincolo
<p>(RV1) Il numero di esemplari gambero allevati in una determinata gamberaia non deve essere mai superiore alla capienza di essa.</p> <p>(RV2) Se in una gamberaia non è presente nessuna tipologia di gambero attualmente in allevamento, non potrà essere effettuata nessuna giornata di prelievo su di essa</p> <p>(RV3) La capacità massima di una gamberaia è di 1 milione di esemplari e deve essere maggiore di 0</p> <p>(RV4) La posizione di una gamberaia (insieme di longitudine e latitudine) deve essere unica</p> <p>(RV5) Una gamberaia può ospitare una tipologia di gambero solo se quest'ultima è presente nell'elenco dei gamberi.</p> <p>(RV6) il codice del codice di una gamberaia deve essere compreso tra 001 e 999</p> <p>(RV7) In una giornata di prelievo deve specificata l'idoneità dei gamberi</p> <p>(RV8) Se durante una giornata di prelievo in una specifica gamberaia l'idoneità dei gamberi allevati non è rispettata, la quantità di gamberi prelevati da quella gamberaia è nulla.</p> <p>(RV9) Una giornata di prelievo può essere effettuata su una gamberaia che è stata registrata nell'elenco delle gamberaie</p> <p>(RV10) In una giornata di prelievo, la quantità di gamberi prelevati in una gamberaia non deve essere mai maggiore alla quantità di gamberi in essa ospitata</p> <p>(RV11) La data fine di un'ospitalità passata in una gamberaia deve essere successiva alla data di inizio di essa.</p> <p>(RV12) La data di inizio di una Ospitalità in una gamberaia, non può essere antecedente alla data di fine dell'ultima ospitalità passata</p> <p>(RV13) La quantità di un'Ospitalità non può essere uguale a 0</p> <p>(RV14) All'interno di una gamberaia il mese di inizio e di fine di un'ospitalità di un gambero deve essere compresa nella stagione di allevamento di quel gambero</p> <p>(RV15) In stagione di allevamento devono essere presente solo nomi di stagioni (estate, primavera, autunno, inverno)</p> <p>(RV16) Il suffisso numerico del codice di un trattamento deve essere compreso tra 001 e 999 La quantità di un prodotto utilizzato in un trattamento deve essere un valore positivo</p> <p>(RV17) L'inserimento di un nuovo trattamento applicato a una gamberaia deve avere necessariamente un orario successivo all'ultimo trattamento applicato nella medesima gamberaia</p> <p>(RV18) L'id deve essere formato un massimo di 4 cifre numeriche</p> <p>(RV19) L'id di un allarme non può essere superiore al numero 9999 e non può essere inferiore a 0</p>

(RV19) L’inserimento di una nuova misurazione di un parametro su una specifica gamberaia deve avere una data o ora successiva all’ultima misurazione del medesimo parametro sulla stessa gamberaia

(RV20) Il consumo energetico di un sensore applicato a gamberaia non può assumere valori negativi

(RV21) Tutte le occorrenze della relazione tra “Gambero” e “Ospitalità Corrente” compaiono anche tra le occorrenze della relazione tra “Gambero” e “Ospitalità”

Tabella 4. Regole di vincolo

Regole di derivazione
<p>(RD1) Il codice della gamberaia è ottenuto da un numero, che deve essere incrementato di uno all’aggiunta di una nuova gamberaia. Es. “GA001”</p> <p>(RD2) Il codice di trattamento è ottenuto aggiungendo al prefisso “T” un numero di tre cifre, che deve essere incrementato di uno all’aggiunta di un nuovo trattamento . Es. “T001”</p> <p>(RD3) La quantità di gamberi ricavata da una gamberaia in una giornata di prelievo è ottenibile considerando la quantità di gamberi presenti in singola ospitalità nella medesima gamberaia, assumendo che l'idoneità dei gamberi in quella ospitalità sia rispettata.</p> <p>(RD4) Per determinare il valoreMisurazione in allarme, è necessario recuperare il valore che non ha rispettato i parametri soglia all'interno dell'entità di misurazione,</p> <p>(RD5) L'ID di un allarme si ottiene aumentando incrementalmente il codice di esso rispetto al codice dell’ultimo allarme</p>

Tabella 5. Regole di derivazione

4. Progettazione Logica

Workpackage	Task	Responsabile
WP2	Progettazione Logica	Foschillo Gennaro

4.1. Ristrutturazione Schema Concettuale

4.1.1. Analisi delle Prestazioni

4.1.1.1. Tavola dei volumi

**Considerazioni effettuate in un periodo di tempo di una stagione (max 6 mesi).

Concetto	Tipo	Volume
Gamberaia	E	50
Gambero	E	20
Ospitalità corrente	E	50
Ospitalità passata	E	50
Applicazione	E	90.000(su ogni gamberaia si effettuano 10 trattamenti al giorno)
Trattamento	E	40
Operatore	E	50
Allarme	E	27.000 (una gamberaia attiva un massimo di 3 allarmi al giorno)
Sensore	E	1000
Sensore Applicato a gamberaia	E	750(15 sensori a gamberaia)
Parametro	E	10
Azienda	E	10
Misurazione	E	6.500.000(considerando tutti i sensori funzionanti aventi un doppio parametro e misurazioni effettuate ogni ora)
Giornata di prelievo	E	6
R8	R	100(considerando che ci possono essere un massimo di 50 ospitalità correnti e i gamberi devono essere allevati per circa 6 mesi minimo)

R9	R	50
R10	R	50
R6	R	90.000
R5	R	90.000(su ogni gamberaia si effettuano 10 applicazioni di trattamento al giorno)
applicato	R	90.000
R1	R	1.000
R2	R	1.500
R0	R	6.500.000 (ogni sensore con due parametri effettua 10 misurazioni al giorno)
R3	R	6.500.000
R4	R	27.000
R5	R	90.000
R6	R	90.000
R7	R	300
R8	R	100
R9	R	50
gestito	R	27.000
installato	R	750

Tabella 6. Tavola dei volumi

4.1.1.2. Tavola delle operazioni

Operazione	Tipo	Frequenza
Operazione 1: Assegnamento una tipologia di gambero in una gamberaia libera	I	40 volte a stagione
Operazione 2 : Assegnamento di un trattamento indicando la gamberaia su cui è associato con relativi dati: inserimento nome dell'attività, codice, materiali utilizzati e la relativa quantità	I	500 volte al giorno
Operazione 3: per ogni gamberaia trova i dati, la tipologia gambero che attualmente alleva e i trattamenti che sono stati applicati ad essi in una specifica stagione	B	50 volte al mese
Operazione 4: Trova tutte le gamberaie che coltivano in quella stagione una particolare tipologia di gambero	I	1 volta a stagione

Operazione 5: memorizza un nuovo sensore alla relativa gamberaia	B	750 volte a stagione
Operazione 6: Salvataggio di una misurazione associato a un parametro di un sensore applicato su una gamberaia	B	24 volte al giorno
Operazione 7: Salvataggio di un nuovo tipo di allarme rilevato e dei suoi dati, inclusa la relativa misurazione	B	150 volte al giorno
Operazione 8: Assegna un operatore per gestire un allarme rilevato e definisci un trattamento relativo	I	150 volte al giorno
Operazione 9: Stampa tutti i dati relativi a una giornata di prelievo, inclusa la quantità totale di gamberi totali prelevati in essa	B	1 volta al mese

Tabella 7. Tavola delle operazioni

4.2. Analisi delle ridondanze

- *Ridondanza 1: l'attributo 'numero Sensori' di una gamberaia è derivabile contando le occorrenze dell'associazione 'installato' a cui tale sensore, applicato alla gamberaia, partecipa*

TIPO: Attributi derivabili da operazioni di conteggio di occorrenze

- *Ridondanza 2: L'attributo 'quantità Totale' nella 'giornata di prelievo' è derivabile dalla somma delle singole quantità nell'entità 'ospitalità corrente' associate alle singole gamberaie collegate a quella giornata di prelievo.*

TIPO: Attributo Derivabile da conteggio di occorrenze

4.2.1. Analisi della ridondanza 1: Numero Sensori

Operazione 5: memorizza un nuovo sensore alla relativa Gamberaia

Con Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Sensore applicato a gamberaia	E	1	S
Installato	R	1	S
Gamberaia	E	1	L
Gamberaia	E	1	S

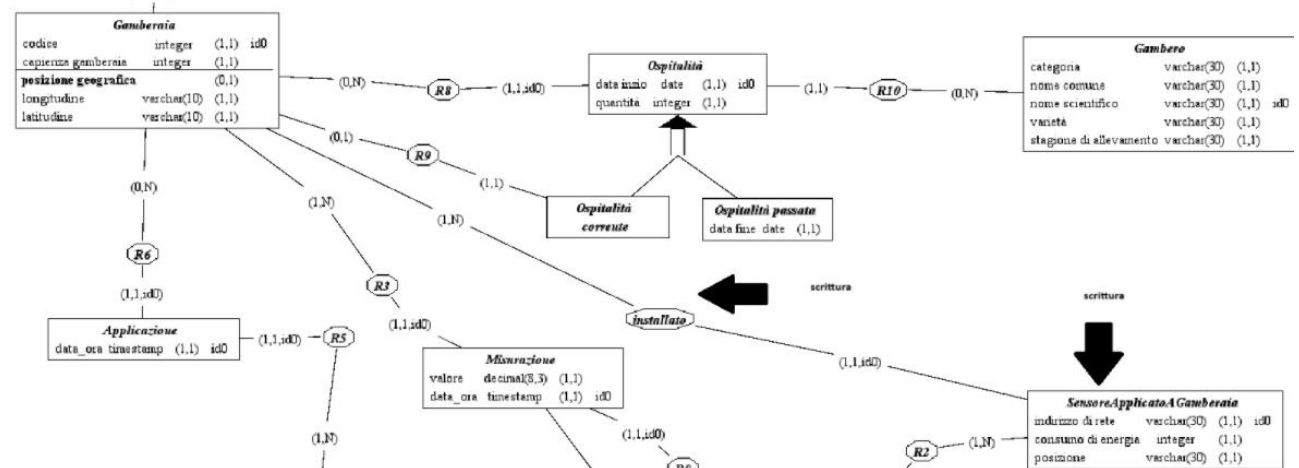
Senza Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Sensore applicato a gamberaia	E	1	S
Installato	R	1	S

4.2.1.1. Valutazione della ridondanza 1

Dopo un'analisi delle operazioni coinvolte e considerando il carico considerato, si nota quanto segue:

- Con la presenza di ridondanza, il costo delle operazioni ammonta a circa 3000 accessi a stagione, di cui 2250 in scrittura e 750 in lettura. Questo valore tiene conto del fatto che, in media, vengono installati 750 sensori in una stagione.
- L'occupazione di memoria è stimata intorno ai 100 byte. È importante notare che il numero di sensori applicati a una gamberaia può essere rappresentato in 4 byte, con un'occupazione massima di 200 byte, nel caso delle 50 gamberaie.
- In assenza di ridondanza, il costo delle operazioni si riduce a 1500 accessi stagionali. Questo perché le operazioni di "Sensore Applicato a Gamberaia" e "Installato" vengono eseguite in scrittura rispettivamente per un massimo di 750 volte a stagione.
- Di conseguenza, si è deciso di rimuovere la ridondanza poiché contribuisce a ridurre il numero complessivo di accessi.



4.2.2. Analisi della ridondanza 2: Quantità Totale in Giornata di prelievo

Operazione 9: Stampa tutti i dati relativi a una giornata di prelievo, inclusa la quantità totale di gamberi totali prelevati in essa

Con Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Giornata di prelievo	E	1	L

Senza Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Giornata di prelievo	E	1	L
R7	R	50	L
Gamberaia	E	1	L
R9	R	50	L
Ospitalità Corrente	E	50	L
Ospitalità	E	50	L

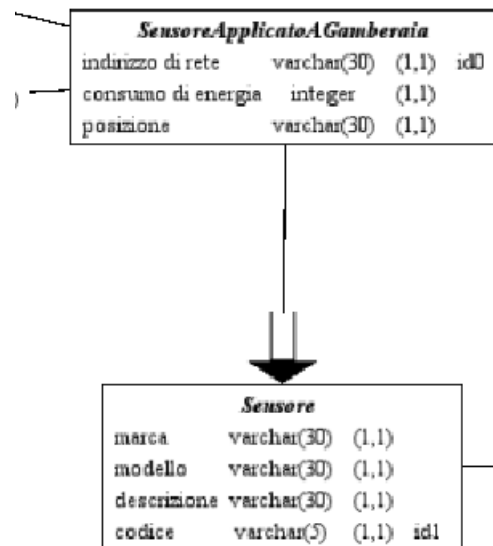
4.2.2.2. Valutazione della ridondanza 2

Dopo aver analizzato le operazioni che coinvolgono la ridondanza si osserva che, con il carico considerato:

- In presenza di ridondanza il costo delle operazioni è di circa 6 accessi a stagione in lettura , considerando che in una stagione vengono effettuate 6 giornate di prelievo.
- L'occupazione di memoria è di circa 48 byte, considerando che la data può essere rappresentata con 4 byte (1 per il giorno, 1 per il mese e 2 per l'anno) e anche la quantità (considerando un massimo di 50.000.000 gamberi nel caso in cui tutte le gamberaie siano coinvolte nel prelievo)
- In assenza di ridondanza il costo delle operazioni è di 1212 accessi stagionali, in quanto le operazioni di lettura nelle entità/associazioni coinvolte vengono effettuate 6 volte a stagione.
- Pertanto, si decide di mantenere la ridondanza in quanto riduce il numero di accessi.

4.3. Eliminazione delle generalizzazioni

4.3.1. Generalizzazione Sensore



Viene applicata la **strategia dell'accorpamento dell'entità figlia della generalizzazione nel genitore**.

Quindi è stata eliminata l'entità "Sensore Applicato a Gamberaia", con i relativi attributi e associazioni incorporati o associati nell'entità padre. È stato aggiunto un attributo booleano "Applicato A Gamberaia" per identificare il sensore e distinguere quelli applicati da quelli non applicati.

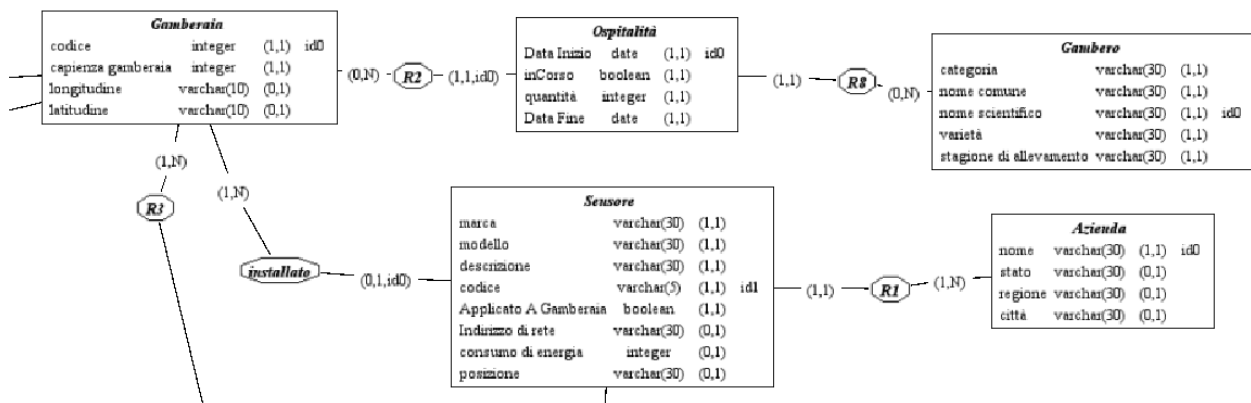
In seguito a questa modifica, è stata introdotta una regola di vincolo

(RV22): se l'attributo "Applicato A Gamberaia" è impostato su false, gli attributi "posizione", "consumo di energia" e "indirizzo di rete" devono essere necessariamente NULL.

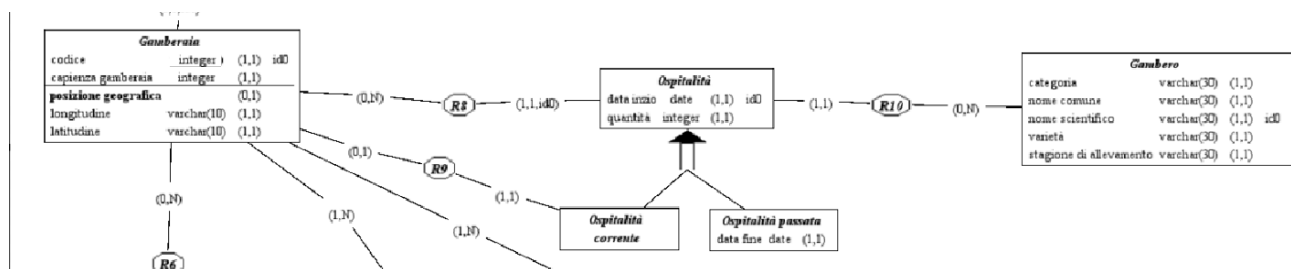
(RV23): solo occorrenze di Sensore con attributo "Applicato A Gamberaia" = TRUE possono essere collegate a "Gamberaia"

Questa strategia è stata adottata nonostante l'incremento di memoria dovuto alla presenza di valori NULL, poiché garantisce un minor numero di accessi rispetto alla soluzione precedente in cui gli attributi sono distribuiti tra entità padre e figlio. Considerando l'utilizzo di 1000 sensori a stagione, di cui 250 non applicati, con la generalizzazione si sarebbero avute 250 operazioni in scrittura di 4 attributi e 750 di 7 attributi. Assegnando un costo di memoria di 4 byte per attributo, il costo totale sarebbe stato di 25 Kbyte. Eliminando la generalizzazione, il costo totale sarebbe aumentato a 32 Kbyte.

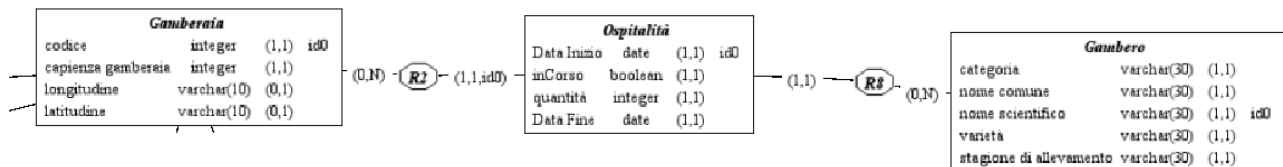
In conclusione, nonostante l'aumento di memoria, l'accorpamento garantisce una migliore efficienza nell'accesso ai dati, rendendo le operazioni di recupero dati meno onerose.



4.3.2. Generalizzazione Ospitalità



Viene applicata la **strategia dell'accorpamento delle entità figlie della generalizzazione nel genitore**.



Le entità "Ospitalità Corrente" e "Ospitalità Passata", insieme ai rispettivi attributi e associazioni, sono state eliminate e incorporate o associate nell'entità padre. È stato aggiunto un attributo booleano di tipo "inCorso", che identifica un'ospitalità attualmente in corso su una Gamberaia se TRUE, altrimenti identifica quell'occorrenza come un'ospitalità passata.

In seguito a questa modifica, sono state introdotte le seguenti regole di vincolo:

(RV24) Ad una Gamberaia può essere associata una sola Ospitalità con attributo "inCorso" = TRUE. Eventualmente, le altre Ospitalità associate alla stessa Gamberaia devono necessariamente avere l'attributo "inCorso" = FALSE e Data Fine diverso da NULL.

(RV25) Se un'occorrenza di tipo "Ospitalità" associata ad una Gamberaia ha attributo "inCorso" = TRUE, l'attributo "Data Fine" è impostato a NULL.

(RV26) Se in "Ospitalità" associata ad una Gamberaia con "inCorso" = TRUE viene aggiunta una "Data Fine", l'attributo "in Corso" sarà impostato a NULL.

(RV27) L'attributo "Data Fine" di due occorrenze successive dell'entità "Ospitalità" deve avere un intervallo cronologico di 6 mesi. Ad esempio, se l'ultima occorrenza di Data Fine è impostata a: 05/01/2024, la successiva deve avere una Data Fine di almeno 6 mesi dopo, quindi: 05/07/2024.

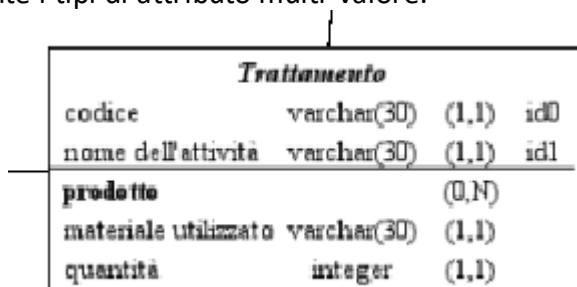
Nonostante l'aumento della occupazione di memoria e lo spreco dovuto alla presenza di valori NULL, questa strategia riduce il numero complessivo di accessi.

4.4. Partizionamento/Accorpamento Entità e Associazioni

4.4.1. Eliminazione attributi multi-valore e composti

4.4.1.1. Partizionamento attributo "prodotto" in "Trattamento"

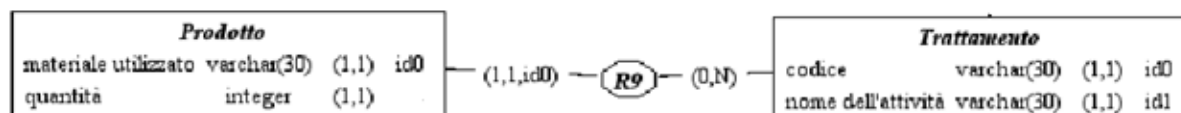
Questa ristrutturazione si rende necessaria perché il modello relazionale non permette di rappresentare direttamente i tipi di attributo multi-valore.



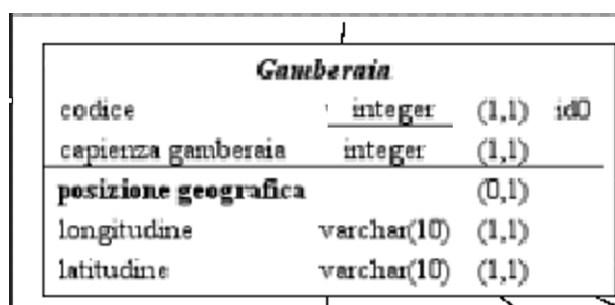
Si effettua un partizionamento in due entità :

"Prodotto", che è deve essere associato ad un solo trattamento .

"Trattamento", che ha tutti gli attributi dell'entità precedente .



4.4.1.2. Eliminazione attributo composto "posizione" in "Gamberaia"



<i>Gamberaia</i>				
codice	integer	(1,1)	id0	
capienza gamberaia	integer	(1,1)		
longitudine	varchar(10)	(0,1)		
latitudine	varchar(10)	(0,1)		

Vincolo aggiuntivo (RV28) : gli attributi “longitudine” e “latitudine” devono essere contemporaneamente NULL oppure definiti.

4.4.1.3. Eliminazione attributo composto “posizione” in “Azienda”

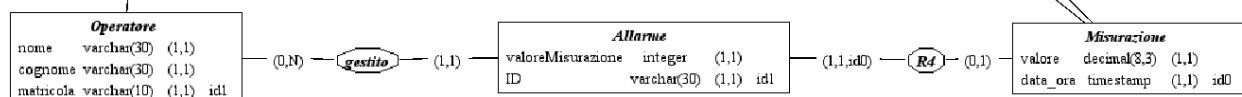
<i>Azienda</i>				
nome	varchar(30)	(1,1)	id0	
posizione		(0,1)		
stato	varchar(30)	(1,1)		
regione	varchar(30)	(1,1)		
città	varchar(30)	(1,1)		

<i>Azienda</i>				
nome	varchar(30)	(1,1)	id0	
stato	varchar(30)	(0,1)		
regione	varchar(30)	(0,1)		
città	varchar(30)	(0,1)		

Vincolo aggiuntivo (RV29) : Gli attributi “stato”, “regione” e “città” devono essere contemporaneamente NULL o definiti.

4.4.2. Accorpamento Entità / Associazioni

4.4.2.1. Accorpamento entità “Allarme” in “Misurazione”

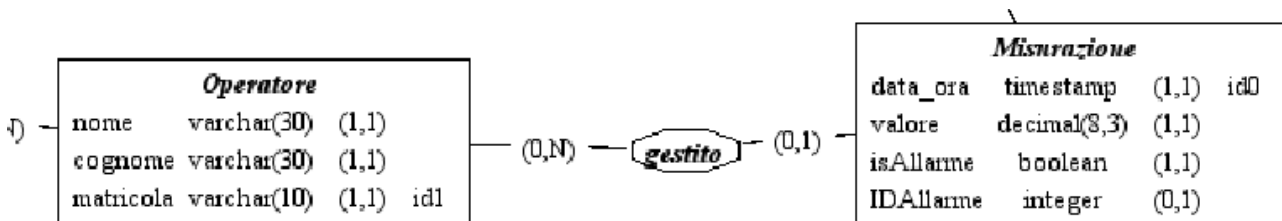


Dopo il raggruppamento delle entità Misurazione ed Allarme in un'unica entità contenente gli attributi di entrambi, viene aggiunto l'attributo booleano di tipo IsAllarme. Questo attributo assume valore TRUE quando il valore della misurazione è superiore alla soglia del parametro associato.

Le operazioni più frequenti in allarme richiedono sempre dati relativi alla misurazione associata. Con questo accorpamento si ottiene un risparmio di accessi necessari per risalire ai dati attraverso l'associazione che li lega. Tuttavia, sarà presente una presenza di valori NULL in IDAllarme, dovuta alla cardinalità (0,1) tra Misurazione ed Allarme.

Per gestire questa situazione, sono definiti i seguenti vincoli:

- (RV26): Se IsAllarme è TRUE, IDAllarme deve essere specificato. Altrimenti IDAllarme è NULL.
- (RV27): Solo misurazioni con "IsAllarme" = TRUE possono essere associate ad un Operatore
- (RV28): Un operatore può gestire solo misurazioni con l'attributo IsAllarme uguale a TRUE.



Scelta degli identificatori principali

GIORNATA DI PRELIEVO -> DATA

GAMBERAIA -> CODICE

OSPITALITA' -> DATA INIZIO

GAMBERO -> NOME SCIENTIFICO

APPLICAZIONE -> DATA_ORA

TRATTAMENTO -> CODICE

PRODOTTO -> NOME DELL'ATTIVITA'

OPERATORE -> MATRICOLA

MISURAZIONE -> DATA_ORA

PARAMETRO -> DESCRIZIONE PARAMETRO

SENSORE -> CODICE

AZIENDA -> NOME

4.5. Schema ristrutturato finale

In seguito vi è la rappresentazione dello schema ristrutturato finale

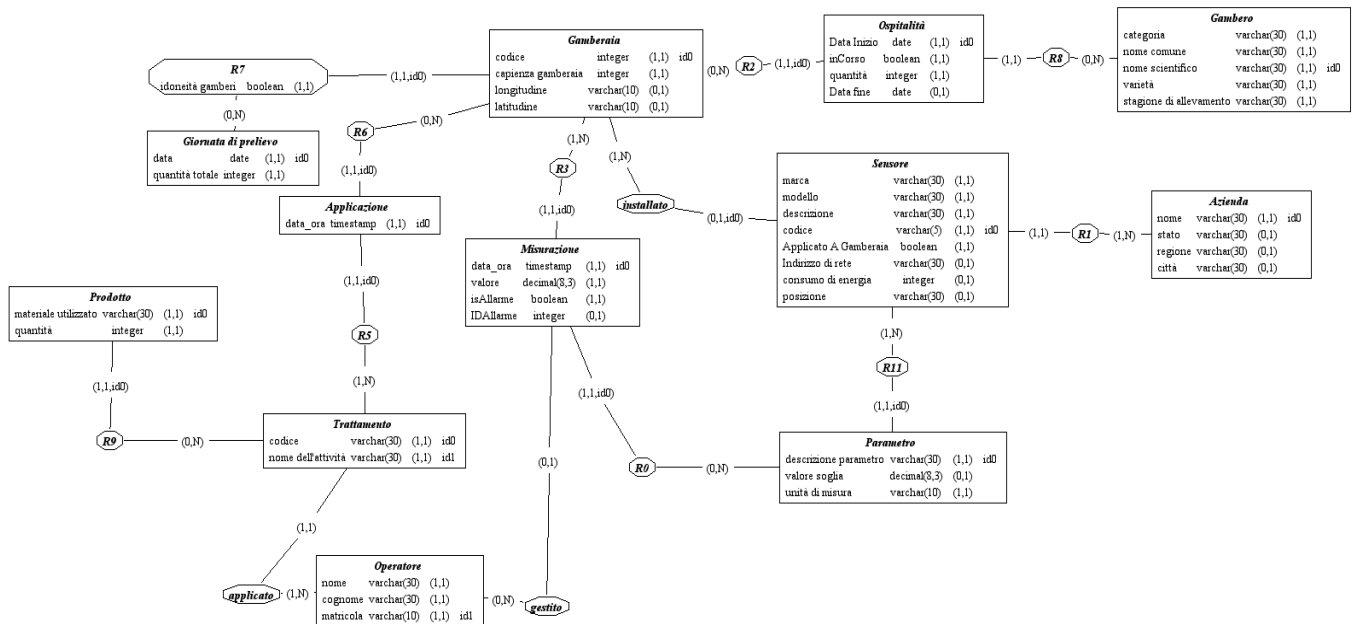


Figura 10. Schema ER Ristrutturato

4.6. Schema logico

A partire da uno schema ristrutturato si costruisce uno schema logico equivalente, che permette di rappresentare le medesime informazioni.

GIORNATA DI PRELIEVO(data, quantità totale)

GAMBERAIA(codice, capienza, longitudine, latitudine, R7Idoneità gamberi, giornata di prelievo)

OSPITALITA'(dataInizio, gamberaia, inCorso, quantità, gambero, dataFine)

GAMBERO(nome scientifico, nome comune, categoria, varietà, stagione di allevamento)

APPLICAZIONE(data, trattamento, gamberaia)

TRATTAMENTO(codice, nome dell'attività)

PRODOTTO(materiale utilizzato, trattamento, quantità)

OPERATORE(matricola, nome, cognome)

MISURAZIONE(data ora, gamberaia, parametro, operatore*, isAllarme, IDAllarme*, valore)

PARAMETRO(descrizione parametro, sensore, valore soglia*, unità di misura)

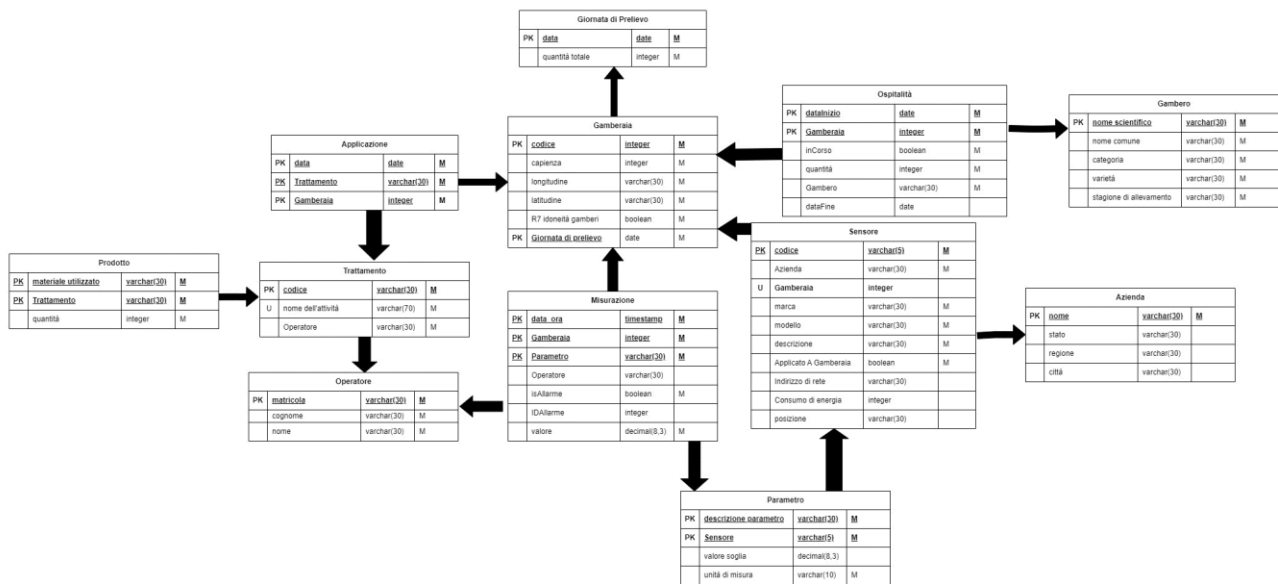
SENSORE(codice, azienda, gamberaia, marca, modello, descrizione, codice, applicato a gamberaia, indirizzo di rete*, consumo di energia*, posizione*)

AZIENDA (nome, stato*, regione*, città*)

___ la sottolineatura indica la PRIMARY KEY della medesima tabella

* gli asterischi indicano la possibilità di avere valori nulli sugli attributi relativi*

Viene inoltre fornita la rappresentazione in tabelle dello schema logico:



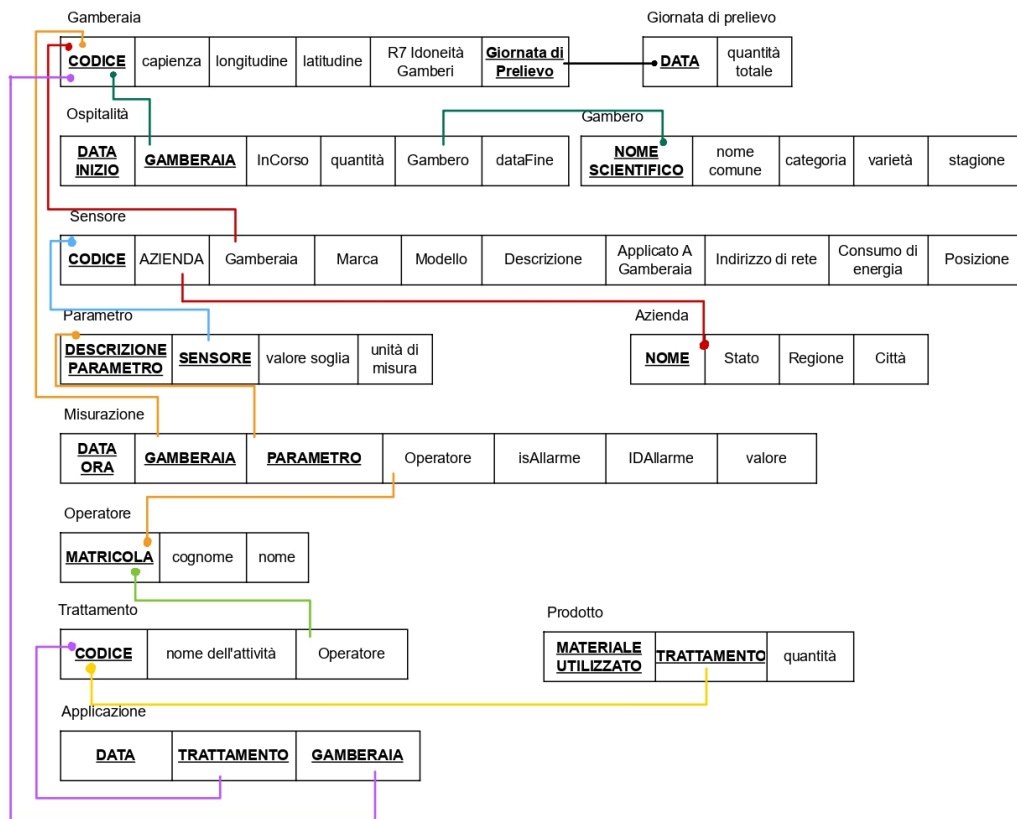
Vincoli aggiuntivi:

Per ogni giornata di prelievo, la quantità di gamberi prelevati da ogni Gamberaia associata ad essa sarà uguale alla somma delle quantità nell'ospitalità della Gamberaia con idoneità = TRUE. Se idoneità = FALSE, alla quantità dei gamberi prelevati sarà aggiunto 0.

Vincoli di integrità referenziale.

.

4.7. Documentazione dello schema logico



In questo schema le frecce rappresentano i vincoli di integrità referenziale.

5. Normalizzazione

Workpackage	Task	Responsabile
WP3	Normalizzazione	Allocca Francesco Vittorio

In questa parte della documentazione si andranno a studiare le proprietà delle forme normali, grazie ad esse si possono certificare la qualità dello schema della nostra basi dati. Se infatti una relazione non soddisfa una forma normale, allora presenta ridondanze e si presta a comportamenti poco desiderabili durante la fase di aggiornamento.

Precisiamo che nella normalizzazione vi sono 4 forme per determinare la qualità del sistema. Una relazione è nella i-sima forma normale se:

- 1- *Prima forma normale (1NF).* Il dominio degli attributi di una relazione deve comprendere solo valori atomici. Quindi non vi devono essere presenti sia attributi multi valore che attributi composti. Nel modello relazionale una medesima relazione dovrebbe essere per definizione in 1NF
- 2- *Seconda forma normale (2NF).* E' in 1NF. Ogni attributo non primo (ovvero che non prende parte alla super chiave della relazione in questione) dipende in maniera piena dalla chiave della medesima relazione
- 3- *Terza forma normale (3NF).* È in seconda forma normale. Per ogni dipendenza funzionale non banale $Y \rightarrow Z$ definita su $R(X)$:
Y è chiave della relazione $R(X)$, oppure, Z è un attributo non primo.
- 4- *Forma normale di Boyce e Codd (BCNF).* E' in prima forma normale. E' in seconda forma normale. Per ogni dipendenza funzionale $Y \rightarrow Z$, non banale definita su $R(X)$
Y contiene la chiave K di $R(X)$ e Y è una super chiave di $R(X)$.

Adesso osserviamo le nostre relazioni e classifichiamo la qualità di esse:

GIORNATA DI PRELIEVO(data, quantità totale)

L'unica DP non banale è: data-> quantità totale. In questo caso rispetta la forma normale di Boyce e Codd

GAMBERAIA(codice, capienza, longitudine, latitudine, R7Idoneità gamberi, giornata di prelievo)

In questo caso non rispetta la *Seconda forma normale*, dato che posso fare come dipendenza funzionale non banale

Codice->longitudine

Codice-> latitudine

Codice-> capienza

In questo caso effettuerò una decomposizione della relazione per normalizzare

Individuiamo le dipendenze funzionali piene non banali.

Mettiamole in forma minimale ed esaminiamo se qualche attributo non primo non dipende in maniera funzionale piena dalla chiave

GAMBERAIA_CARATTERISTICHE(codice, capienza, longitudine, latitudine)

Codice->longitudine

Codice-> latitudine

Codice-> capienza

PRELIEVO (codice, R7Idoneità gamberi, giornata di prelievo)

Codice, giornata di prelievo -> R7Idoneità gamberi

Analizzando se la decomposizione è una decomposizione senza perdite,

-Sia R una relazione su PRELIEVO

$GAMBERAIA_CARATTERISTICHE \subset GAMBERAIA$ e $PRELIEVO \subset GAMBERAIA$

-R1(GAMBERAIA_CARATTERISTICHE) proiezione di R su GAMBERAIA_CARATTERISTICHE

-R2(PRELIEVO) proiezione di R su PRELIEVO

Sia $X0 = GAMBERAIA_CARATTERISTICHE \cap PRELIEVO$ diverso dall'insieme vuoto.

$GAMBERAIA_CARATTERISTICHE \cup PRELIEVO = PRELIEVO$

Allora possiamo accertare che non ci saranno tuple spurie ed questa decomposizione è decomposizione senza perdite

Inoltre ciascuna delle dipendenze funzionali dello schema originario coinvolge attributi che compaiono tutti insieme in uno degli schemi decomposti

In questo modo possiamo notare che sia GAMBERAIA_CARATTERISTICHE che PRELIEVO, rispettano la forma normale di Boyce e Codd.

Ovviamente di dovranno essere dei vincoli di integrità referenziale sia a codice che a giornata di prelievo

OSPITALITA'(dataInizio, gamberaia, inCorso, quantità, gambero, dataFine)

Le dipendenze funzionali non banali di questa entità sono

dataInizio , gamberaia->inCorso

dataInizio , gamberaia->quantità

dataInizio , gamberaia->gambero

dataInizio , gamberaia->dataFine

ci deve essere sempre il vincolo d'integrità referenziale a Gamberaia attraverso il codice di essa

in questo caso si rispetta le regole per la forma normale di Boyce e Codd.

APPLICAZIONE(data, trattamento, gamberaia)

Quando abbiamo una relazione in cui ogni attributo è una chiave e tutti insieme formano la chiave primaria, allora quella relazione è in Forma Normale di Boyce-Codd (BCNF)

TRATTAMENTO(codice, nome dell'attività, operatore)

Ad un trattamento è sempre associato lo stesso operatore, ed un operatore può effettuare più trattamenti

Le dipendenze funzionali non banali di questa entità sono

Codice-> nome dell'attività

Codice->operatore

In questo caso rispetta la Forma Normale di Boyce-Codd.

Ci deve essere sempre il vincolo d'integrità referenziale a operatore attraverso il la matricola nella rispettiva tabella.

PRODOTTO(materiale utilizzato, trattamento, quantità)

Ogni trattamento ha un'unica quantità di prodotto associata ad esso

Non ci sono situazioni in cui un trattamento può avere più di una quantità di prodotto associata ad esso

Le dipendenze funzionali non banali di questa entità sono:

materiale utilizzato, trattamento->quantità

Possiamo dire che la relazione è in Forma Normale di Boyce-Codd (BCNF), poiché ogni dipendenza funzionale non banale è determinata dalla super chiave

OPERATORE(matricola, nome, cognome)

Le dipendenze funzionali non banali di questa entità sono:

Matricola->nome

Matricola->cognome

In altre parole, dato un valore di matricola, è possibile determinare univocamente il nome e il cognome dell'operatore. Questo tipo di dipendenza funzionale è una caratteristica della Forma Normale di Boyce-Codd

MISURAZIONE(data ora, gamberaia, parametro, operatore*, isAllarme, IDAllarme*, valore)

Significa che ogni tupla nella tabella MISURAZIONE è univocamente identificata da una combinazione di valori in questi tre campi. Questo suggerisce che ogni possibile combinazione di valori per questi tre campi identifica una singola misurazione, senza ripetizioni.

Dato che per ottenere il valore di una misurazione dovrai sapere su quale gamberaia è stata effettuata, da quale parametro deriva e in che data è stata fatta, ricavato il valore di esso andrai a settare i rimanenti attributi

data_ora, gamberaia, parametro -> operatore
 data_ora, gamberaia, parametro -> isAllarme
 data_ora, gamberaia, parametro -> IDAllarme
 data_ora, gamberaia, parametro -> valore

la tabella MISURAZIONE soddisfa i requisiti della forma normale di Boyce-Codd.

PARAMETRO(descrizione parametro, sensore, valore soglia*, unità di misura)

DESCRIZIONE PARAMETRO	SENSORE	VALORE SOGLIA	UNITÀ DI MISURA
Temperatura acqua	S001	25	°C
Temperatura acqua	S002	25	°C
Rilevazione PH	S001	12	Ph
Calcolo salinità	S003	0.45	mS\cm

In questo caso non rispetta la *Seconda forma normale*, dato che posso fare come dipendenza funzionale non banale

descrizione parametro-> valore soglia
 descrizione parametro-> unità di misura

In questo caso effettuerò una decomposizione della relazione per normalizzare

Quindi in questo caso avresti

PARAMETRO(descrizione parametro, valore soglia*, unità di misura)
CARATTERISTICHE(descrizione parametro, sensore)

Ovviamente deve esserci un vincolo d'integrità referenziale a CARATTERISTICHE sull'attributo descrizione parametro nella tabella Parametro

Un ulteriore vincolo d'integrità referenziale in CARATTERISTICHE sull'attributo sensore che deve essere presente come codice nella tabella Sensore

Analizzando la nuova decomposizione:

Sia R una relazione su PARAMETRO.

$PARAMETRO_DETTAGLI \subset PARAMETRO$ e $CARATTERISTICHE \subset PARAMETRO$.

$R1(PARAMETRO_DETTAGLI)$ è la proiezione di R su $PARAMETRO_DETTAGLI$.

$R2(CARATTERISTICHE)$ è la proiezione di R su $CARATTERISTICHE$.

Sia $X0 = PARAMETRO_DETTAGLI \cap CARATTERISTICHE$ diverso dall'insieme vuoto.

$PARAMETRO_DETTAGLI \cup CARATTERISTICHE = PARAMETRO$.

Allora possiamo accertare che non ci saranno tuple spurie e questa decomposizione è una decomposizione senza perdite.

Inoltre, ciascuna delle dipendenze funzionali dello schema originario coinvolge attributi che compaiono tutti insieme in uno degli schemi decomposti.

In $PARAMETRO_DETTAGLI$

descrizione parametro \rightarrow valore soglia

descrizione parametro \rightarrow unità di misura

In questo modo possiamo notare che sia $PARAMETRO_DETTAGLI$ che $CARATTERISTICHE$ rispettano la forma normale di Boyce e Codd.

$SENSORE(\underline{\text{codice}}, \text{azienda}, \text{gamberaia}, \text{marca}, \text{modello}, \text{descrizione}, \text{applicato a gamberaia}, \text{indirizzo di rete}^*, \text{consumo di energia}^*, \text{posizione}^*)$

Le dipendenze funzionali non banali di questa entità sono:

Codice \rightarrow Azienda

Codice \rightarrow gamberaia

Codice \rightarrow marca

Codice \rightarrow modello

Codice \rightarrow descrizione

Codice \rightarrow applicato a gamberaia

Codice \rightarrow indirizzo di rete

Codice \rightarrow consumo di energia

Codice \rightarrow posizione

Queste dipendenze funzionali implicano che il codice del sensore è la chiave primaria e che tutte le altre proprietà dipendono direttamente da essa.

Ogni dipendenza funzionale non banale deve essere una dipendenza funzionale completa e diretta dalla chiave primaria. Poiché tutte le dipendenze funzionali menzionate soddisfano questa condizione, possiamo concludere che lo schema dell'entità "SENSORE" è in BCNF.

AZIENDA (nome, stato*, regione*, città*)

Le dipendenze funzionali non banali di questa entità sono:

nome->stato

nome->regione

nome->città

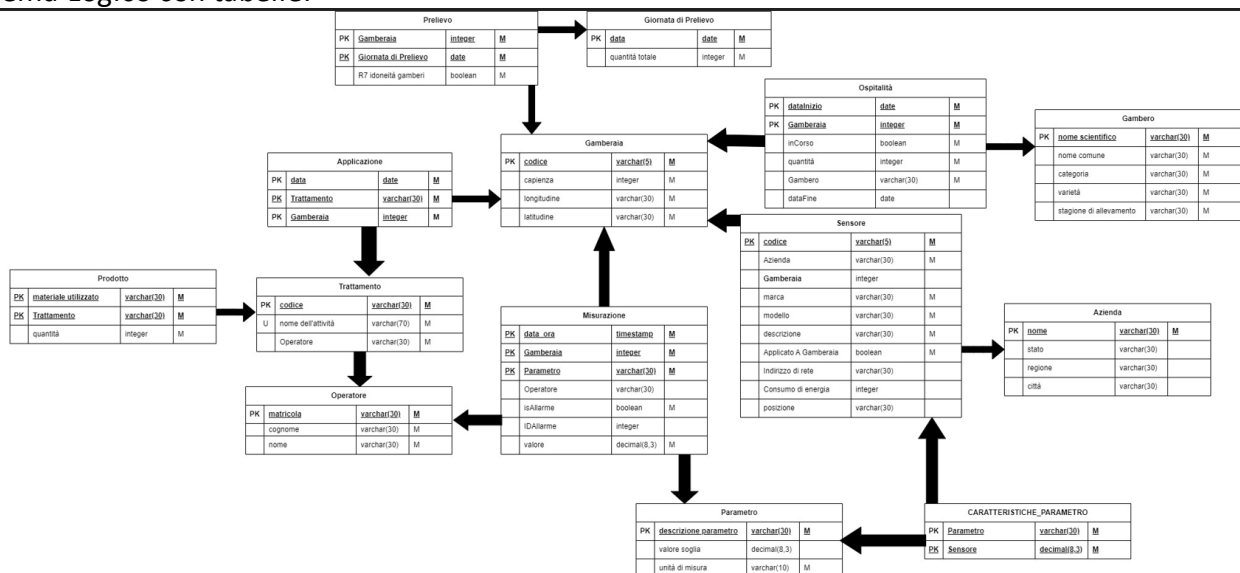
Tutte le dipendenze funzionali nell'entità "Azienda" sono direttamente e completamente determinate dalla chiave primaria, il nome dell'azienda.

Poiché tutte le dipendenze funzionali soddisfano questa condizione, possiamo affermare che lo schema dell'entità "Azienda" è conforme alla forma normale di Boyce-Codd (BCNF).

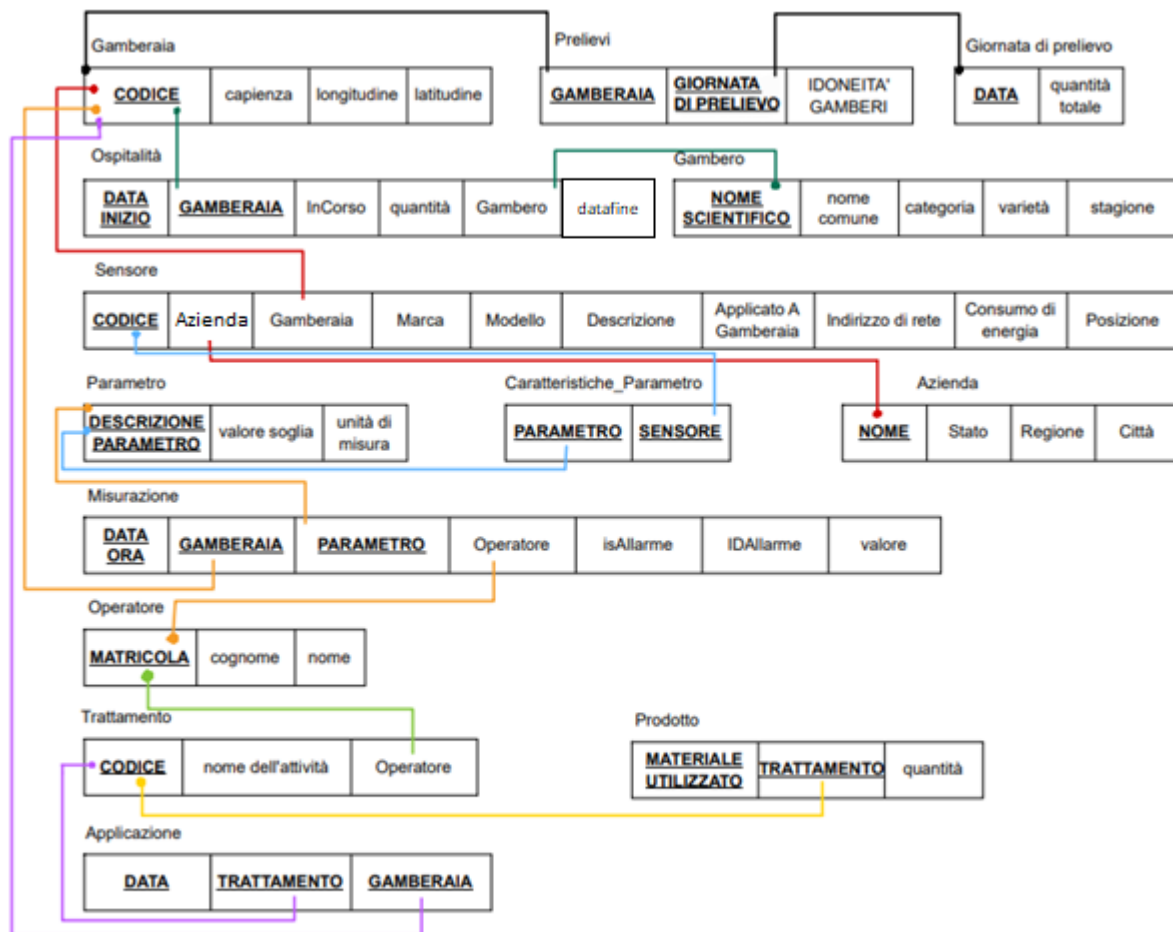
Se una relazione rispetta la forma normale di Boyce-Codd (BCNF), allora automaticamente rispetta anche la 1NF, la 2NF e la 3NF. Questo perché la BCNF è una forma più forte delle precedenti normalizzazioni.

Tutte le relazioni sopra citate, rispettano la BCNF, allora possiamo essere certi che il database è stato progettato in modo efficiente e che non esistono problemi di ridondanza, anomalie di aggiornamento o dipendenze funzionali indesiderate.

Schema Logico con tabelle:



Schema logico normalizzato con vincoli di integrità referenziale:



Script Creazione e Popolamento Database

Workpackage	Task	Responsabile
WP2	SQL: Script creazione e popolamento	Foschillo Gennaro

```
CREATE TABLE Gamberaia (
  codice INTEGER PRIMARY KEY,
  capienza INTEGER CHECK (capienza >= 0 AND capienza <= 1000000),
  longitudine VARCHAR(10) ,
  latitudine VARCHAR(10),
  UNIQUE(longitudine , latitudine),
  CONSTRAINT chk_codice CHECK (codice <= 999 AND codice > 0),
  CONSTRAINT chk_lat_long_null CHECK (
    (latitudine IS NULL AND longitudine IS NULL) OR
    (latitudine IS NOT NULL AND longitudine IS NOT NULL)
  )
);
```

```
INSERT INTO Gamberaia (codice, capienza, longitudine, latitudine)
VALUES
(1, 100000, '43.123456', '12.345678'),
(2, 200000, '42.987654', '13.456789'),
(3, 300000, '41.111111', '14.567890'),
(4, 400000, '40.999999', '15.678901'),
(5, 500000, '43.555555', '12.111111'),
(6, 600000, '42.222222', '13.222222'),
(7, 700000, '41.333333', '14.333333');
```

```
INSERT INTO Gamberaia (codice, capienza, longitudine, latitudine)
VALUES
(8, 800000, '40.666666', '15.444444'),
(9, 900000, '43.777777', '12.555555'),
(10, 1000000, '42.888888', '13.666666');
```

```
CREATE TABLE gambero (
  nome_scientifico VARCHAR(30) PRIMARY KEY,
  categoria VARCHAR(30) NOT NULL,
  nome_comune VARCHAR(30) NOT NULL,
  varietà VARCHAR(30) NOT NULL,
  stagione_allevamento VARCHAR(30) NOT NULL,
  CONSTRAINT chk_stagione_allevamento CHECK (stagione_allevamento IN ('primavera', 'estate',
'autunno', 'inverno'))
);
```

```
INSERT INTO gambero (nome_scientifico, categoria, nome_comune, varietà,
stagione_allevamento)
VALUES
```

```
( 'Aristaeomorpha foliacea', 'Gamberi Giganti', 'Gambero Rosso', 'Gamberi Rossi del Mediterraneo',
'estate'),
('Nephrops norvegicus', 'Gamberi Giganti', 'Gambero Nordico', 'Gambero Nordico', 'estate'),
('Pandalus borealis', 'Gamberi Nordici', 'Gambero Artico', 'Gambero Artico', 'inverno'),
('Penaeus monodon', 'Gamberi Tropicali', 'Gambero Tiger', 'Gambero Tiger', 'primavera'),
('Penaeus vannamei', 'Gamberi Tropicali', 'Gambero Bianco', 'Gambero Bianco', 'estate'),
('Penaeus chinensis', 'Gamberi Tropicali', 'Gambero Cinese', 'Gambero Cinese', 'estate'),
('Penaeus indicus', 'Gamberi Tropicali', 'Gambero Indiano', 'Gambero Indiano', 'primavera'),
('Penaeus merguensis', 'Gamberi Tropicali', 'Gambero Merguensis', 'Gambero Merguensis',
'estate'),
('Penaeus semisulcatus', 'Gamberi Tropicali', 'Gambero Semisulcatus', 'Gambero Semisulcatus',
'autunno');
```

```
CREATE TABLE azienda (
    nome VARCHAR(30) PRIMARY KEY,
    stato VARCHAR(30),
    regione VARCHAR(30),
    città VARCHAR(30),
    CONSTRAINT check_ubicazione CHECK (
        (stato IS NOT NULL AND regione IS NOT NULL AND città IS NOT NULL) OR
        (stato IS NULL AND regione IS NULL AND città IS NULL)
    )
);
```

```
INSERT INTO azienda (nome, stato, regione, città) VALUES
('Acme Inc.', 'USA', 'California', 'Los Angeles'),
('Global Enterprises', 'UK', 'England', 'London'),
('Tech Solutions', 'Canada', 'Ontario', 'Toronto'),
('BioTech Industries', 'Germany', 'Bavaria', 'Munich'),
('Eco Innovations', 'France', 'Île-de-France', 'Paris'),
('Green Energy', 'Italy', 'Lombardia', 'Milano'),
('Smart Systems', 'Spain', 'Catalonia', 'Barcelona'),
('Pacific Corp.', 'Australia', 'New South Wales', 'Sydney'),
('Asia Ventures', 'Japan', 'Tokyo', 'Tokyo');
INSERT INTO azienda (nome) VALUES
('Latin Resources');
```

```
CREATE TABLE operatore (
    matricola VARCHAR(30) PRIMARY KEY,
    cognome VARCHAR(30) NOT NULL,
    nome VARCHAR(30) NOT NULL
);
```

```
INSERT INTO operatore (matricola, cognome, nome)
VALUES
('00003', 'Verdi', 'Luca'),
('00004', 'Neri', 'Anna'),
('00005', 'Giallo', 'Giovanni');
```

```
('00006', 'Blu', 'Laura'),  
('00007', 'Verde', 'Paolo'),  
('00008', 'Rosa', 'Sara'),  
('00009', 'Bianco', 'Marco'),  
('00010', 'Nera', 'Giulia');
```

```
CREATE TABLE trattamento (  
    codice VARCHAR(30) PRIMARY KEY,  
    nome_attivita VARCHAR(30) UNIQUE NOT NULL,  
    operatore VARCHAR(30) NOT NULL,  
    CONSTRAINT fk_operatore_trattamento FOREIGN KEY (operatore) REFERENCES  
operatore(matricola),  
    CONSTRAINT chk_codice_format CHECK (codice ~ '^T[0-9]{3}$')  
);
```

```
INSERT INTO trattamento (codice, nome_attivita, operatore)  
VALUES  
('T001', 'Pulizia vasche', '00003'),  
('T002', 'Alimentazione gamberi', '00004'),  
('T003', 'Controllo acqua', '00005'),  
('T004', 'Riproduzione gamberi', '00006'),  
('T005', 'Pulizia filtri', '00007'),  
('T006', 'Monitoraggio temperatura', '00008'),  
('T007', 'Raccolta gamberi', '00009'),  
('T008', 'Sostituzione acqua', '00004'),  
('T009', 'Analisi acqua', '00005'),  
('T010', 'Manutenzione impianti', '00003'),  
('T011', 'Pulizia Gamberi', '00010');
```

```
CREATE TABLE Prodotto (  
    materiale_utilizzato VARCHAR(30),  
    trattamento VARCHAR(30),  
    quantita INTEGER NOT NULL CHECK (quantita >= 0),  
    PRIMARY KEY (materiale_utilizzato, trattamento),  
    FOREIGN KEY (trattamento) REFERENCES Trattamento(codice)  
);
```

```
INSERT INTO Prodotto (materiale_utilizzato, trattamento, quantita)  
VALUES  
('Detergente speciale', 'T001', 500),  
('Alimento per gamberi', 'T002', 1000),  
('Reagente di controllo', 'T003', 200),  
('Sostanza riproduttiva', 'T004', 300),  
('Filtro sostitutivo', 'T005', 400),  
('Sonda di temperatura', 'T006', 150),  
('Reti di raccolta', 'T007', 250),  
('Acqua di sostituzione', 'T008', 1000),
```

```
('Kit di analisi', 'T009', 50),  
('Pulitore per gamberi', 'T011', 200);
```

```
CREATE TABLE Parametro (  
    descrizione_parametro VARCHAR(30) PRIMARY KEY,  
    valore_soglia DECIMAL(8, 3),  
    unita_misura VARCHAR(10) NOT NULL  
);
```

```
INSERT INTO Parametro (descrizione_parametro, valore_soglia, unita_misura)  
VALUES  
('Temperatura acqua', 25.0, '°C'),  
('pH acqua', 7.5, 'pH'),  
('Ossigeno disciolto', 5.0, 'mg/L'),  
('Nitrogeno ammoniacale', 0.5, 'mg/L'),  
('Densità gamberi', 500.0, 'g/m3');
```

```
CREATE OR REPLACE FUNCTION check_inCorso_single_occurrence(p_gamberaia INTEGER,  
p_data_inizio DATE)  
RETURNS BOOLEAN  
AS $$  
BEGIN  
    IF EXISTS (  
        SELECT 1 FROM OSPITALITA  
        WHERE Gamberaia = p_gamberaia AND inCorso = TRUE AND data_inizio != p_data_inizio  
    ) THEN  
        RETURN FALSE;  
    ELSE  
        RETURN TRUE;  
    END IF;  
END;  
$$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE FUNCTION check_stagione_allevamento(p_data_inizio DATE, p_gambero  
VARCHAR(30))  
RETURNS BOOLEAN  
AS $$  
BEGIN  
    RETURN EXISTS (  
        SELECT 1  
        FROM Gambero  
        WHERE nome_scientifico = p_gambero AND  
        (  
            (stagione_allevamento = 'primavera' AND EXTRACT(MONTH FROM p_data_inizio) BETWEEN  
3 AND 5) OR
```

```

        (stagione_allevamento = 'estate' AND EXTRACT(MONTH FROM p_data_inizio) BETWEEN 6
AND 8) OR
        (stagione_allevamento = 'autunno' AND EXTRACT(MONTH FROM p_data_inizio) BETWEEN 9
AND 11) OR
        (stagione_allevamento = 'inverno' AND EXTRACT(MONTH FROM p_data_inizio) IN (12, 1, 2))
    )
);
END;
$$ LANGUAGE plpgsql;

```

```

CREATE OR REPLACE FUNCTION check_quantità_not_exceed_capienza(p_gamberaia INTEGER,
p_quantità INTEGER)
RETURNS BOOLEAN
AS $$
BEGIN
    RETURN p_quantità <= (SELECT capienza FROM Gamberaia WHERE codice = p_gamberaia);
END;
$$ LANGUAGE plpgsql;

```

```

CREATE TABLE OSPITALITA (
    data_inizio DATE,
    Gamberaia INTEGER,
    inCorso BOOLEAN,
    quantità INTEGER CHECK (quantità > 0),
    Gambero VARCHAR(30),
    data_fine DATE DEFAULT NULL ,
    PRIMARY KEY (data_inizio, Gamberaia),
    FOREIGN KEY (Gamberaia) REFERENCES Gamberaia(codice) ON DELETE CASCADE,
    FOREIGN KEY (Gambero) REFERENCES Gambero(nome_scientifico),
    CONSTRAINT check_data_fine_after_data_inizio CHECK (
        data_fine IS NULL OR data_fine >= data_inizio
    ),
    CONSTRAINT check_inCorso_single_occurrence CHECK (
        (inCorso = TRUE AND check_inCorso_single_occurrence(Gamberaia, data_inizio) = TRUE) OR
        (inCorso = FALSE)
    ),
    CONSTRAINT check_data_fine CHECK (
        (inCorso = TRUE AND data_fine IS NULL) OR (inCorso = FALSE AND data_fine IS NOT NULL)
    ),
    CONSTRAINT check_stagione_allevamento CHECK (
        check_stagione_allevamento(data_inizio, Gambero)
    ),
    CONSTRAINT check_quantità_not_exceed_capienza CHECK (
        check_quantità_not_exceed_capienza(Gamberaia, quantità)
    )
);

```

```
INSERT INTO ospitalita (data_inizio, Gamberaia, inCorso, quantità, Gambero, data_fine)
VALUES
('2022-06-01', 1, TRUE, 5000, 'Aristaeomorpha foliacea', NULL),
('2022-07-01', 2, TRUE, 3000, 'Nephrops norvegicus', NULL),
('2022-01-01', 3, FALSE, 2000, 'Pandalus borealis', '2022-02-28'),
('2022-03-01', 4, TRUE, 4000, 'Penaeus monodon', NULL),
('2022-08-01', 5, TRUE, 6000, 'Penaeus vannamei', NULL);
```

```
CREATE OR REPLACE FUNCTION check_ospitalita_in_corso() RETURNS BOOLEAN AS $$
BEGIN
    RETURN EXISTS (
        SELECT 1
        FROM Ospitalita
        WHERE incorso = TRUE
    );
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TABLE Giornata_di_Prelievo (
    data DATE PRIMARY KEY,
    quantità_totale INT NOT NULL DEFAULT 0,
    CONSTRAINT check_ospitalita_in_corso CHECK (
        check_ospitalita_in_corso()
    )
);
```

```
INSERT INTO Giornata_di_prelievo (data)
VALUES ('2022-08-21');
INSERT INTO Giornata_di_prelievo (data)
VALUES ('2022-07-30');
```

```
CREATE OR REPLACE FUNCTION check_gamberaia_in_corso(codice_gamberaia INTEGER) RETURNS
BOOLEAN AS $$
BEGIN
    RETURN EXISTS (
        SELECT 1
        FROM Ospitalita
        WHERE Gamberaia = codice_gamberaia AND inCorso = TRUE
    );
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TABLE Prelievo (
    codice_gamberaia INTEGER,
    data DATE NOT NULL,
```

```
idoneità_gamberi BOOLEAN NOT NULL,  
PRIMARY KEY (codice_gamberaia, data),  
FOREIGN KEY (codice_gamberaia) REFERENCES Gamberaia(codice),  
FOREIGN KEY (data) REFERENCES Giornata_di_prelievo(data),  
CONSTRAINT fk_gamberaia_inCorso  
    CHECK (check_gamberaia_in_corso(codice_gamberaia))  
);
```

```
INSERT INTO prelievo (codice_gamberaia, data,idoneità_gamberi) VALUES (2 , '2022-08-21' ,  
TRUE);
```

```
CREATE OR REPLACE FUNCTION check_data_successiva(a_gamberaia INTEGER, a_data  
TIMESTAMP)  
RETURNS BOOLEAN  
AS $$  
BEGIN  
    RETURN a_data > COALESCE((SELECT MAX(data) FROM applicazione WHERE a_gamberaia =  
gamberaia), 'epoch'::timestamp);  
END;  
$$ LANGUAGE plpgsql;
```

```
CREATE TABLE applicazione (  
    data TIMESTAMP,  
    trattamento VARCHAR(30),  
    gamberaia INTEGER,  
    PRIMARY KEY (data, gamberaia, trattamento),  
    FOREIGN KEY (trattamento) REFERENCES trattamento(codice),  
    FOREIGN KEY (gamberaia) REFERENCES gamberaia(codice),  
    CONSTRAINT check_data_successiva CHECK (  
        check_data_successiva(gamberaia, data)  
    )  
);
```

```
INSERT INTO applicazione (data, trattamento, gamberaia)  
VALUES  
( '2022-01-01 10:00:00', 'T001', 1),  
( '2022-01-02 11:00:00', 'T002', 2),  
( '2022-01-03 12:00:00', 'T003', 3),  
( '2022-01-04 13:00:00', 'T004', 4),  
( '2022-01-05 14:00:00', 'T005', 5),  
( '2022-01-06 15:00:00', 'T006', 6),  
( '2022-01-07 16:00:00', 'T007', 7),  
( '2022-01-08 17:00:00', 'T008', 8),  
( '2022-01-09 18:00:00', 'T009', 9),  
( '2022-01-10 19:00:00', 'T010', 10),  
( '2022-01-11 20:00:00', 'T011', 1);
```

```

CREATE TABLE sensore (
  codice_sensore VARCHAR(5) PRIMARY KEY,
  gameraia INTEGER ,
  azienda VARCHAR(30) NOT NULL,
  modello VARCHAR(30) NOT NULL,
  descrizione VARCHAR(30) NOT NULL,
  marca VARCHAR(30) NOT NULL,
  applicato_a_gameraia BOOLEAN DEFAULT FALSE ,
  indirizzo_rete VARCHAR(30),
  consumo_energia INTEGER CHECK (
    (applicato_a_gameraia = TRUE AND consumo_energia >= 0) OR (applicato_a_gameraia = FALSE
    AND consumo_energia = NULL)
  ) ,
  posizione VARCHAR(30),
  FOREIGN KEY (gameraia) REFERENCES gameraia(codice) ON DELETE CASCADE,
  FOREIGN KEY (azienda) REFERENCES azienda(nome),
  CONSTRAINT check_applicato_a_gameraia CHECK (
    (gameraia IS NULL AND applicato_a_gameraia = FALSE) OR
    (gameraia IS NOT NULL AND applicato_a_gameraia = TRUE)
  ),
  CONSTRAINT check_null_fields CHECK (
    (gameraia IS NULL AND indirizzo_rete IS NULL AND consumo_energia IS NULL AND posizione
    IS NULL) OR
    (gameraia IS NOT NULL AND (indirizzo_rete IS NOT NULL OR consumo_energia IS NOT NULL
    OR posizione IS NOT NULL))
  )
);

```

```

INSERT INTO sensore (codice_sensore, azienda, modello, descrizione, marca)
VALUES
('S001', 'Acme Inc.', 'Modello 1', 'Descrizione 1', 'Marca 1'),
('S002', 'Global Enterprises', 'Modello 2', 'Descrizione 2', 'Marca 2'),
('S003', 'Tech Solutions', 'Modello 3', 'Descrizione 3', 'Marca 3'),
('S004', 'BioTech Industries', 'Modello 4', 'Descrizione 4', 'Marca 4'),
('S005', 'Eco Innovations', 'Modello 5', 'Descrizione 5', 'Marca 5');

```

```

INSERT INTO sensore (codice_sensore, gameraia, azienda, modello, descrizione, marca,
applicato_a_gameraia, indirizzo_rete, consumo_energia, posizione)
VALUES
('S006', 1, 'Acme Inc.', 'Modello 6', 'Descrizione 6', 'Marca 6', TRUE, '192.168.1.1', 10, 'Posizione 1'),
('S007', 2, 'Global Enterprises', 'Modello 7', 'Descrizione 7', 'Marca 7', TRUE, '192.168.1.2', 20,
'Posizione 2'),
('S008', 3, 'Tech Solutions', 'Modello 8', 'Descrizione 8', 'Marca 8', TRUE, '192.168.1.3', 30,
'Posizione 3'),
('S009', 4, 'BioTech Industries', 'Modello 9', 'Descrizione 9', 'Marca 9', TRUE, '192.168.1.4', 40,
'Posizione 4'),

```



```
('S010', 5, 'Eco Innovations', 'Modello 10', 'Descrizione 10', 'Marca 10', TRUE, '192.168.1.5', 50,
'Posizione 5'),
('S011', 6, 'Green Energy', 'Modello 11', 'Descrizione 11', 'Marca 11', TRUE, '192.168.1.6', 60,
'Posizione 6'),
('S012', 7, 'Smart Systems', 'Modello 12', 'Descrizione 12', 'Marca 12', TRUE, '192.168.1.7', 70,
'Posizione 7'),
('S013', 8, 'Pacific Corp.', 'Modello 13', 'Descrizione 13', 'Marca 13', TRUE, '192.168.1.8', 80,
'Posizione 8'),
('S014', 9, 'Asia Ventures', 'Modello 14', 'Descrizione 14', 'Marca 14', TRUE, '192.168.1.9', 90,
'Posizione 9'),
('S015', 10, 'Latin Resources', 'Modello 15', 'Descrizione 15', 'Marca 15', TRUE, '192.168.1.10', 100,
'Posizione 10');
```

```
CREATE TABLE caratteristiche_parametro(
    sensore VARCHAR(5),
    parametro VARCHAR(30),
    PRIMARY KEY (sensore, parametro),
    CONSTRAINT fk_sensore FOREIGN KEY (sensore) REFERENCES sensore(codice_sensore ),
    CONSTRAINT fk_parametro FOREIGN KEY (parametro) REFERENCES
parametro(descrizione_parametro)
);
```

```
INSERT INTO caratteristiche_parametro (sensore, parametro)
VALUES
('S001', 'Temperatura acqua'),
('S001', 'pH acqua'),
('S002', 'Ossigeno disciolto'),
('S002', 'Nitrogeno ammoniacale'),
('S003', 'Densità gamberi'),
('S003', 'Temperatura acqua'),
('S004', 'pH acqua'),
('S004', 'Ossigeno disciolto'),
('S005', 'Nitrogeno ammoniacale'),
('S005', 'Densità gamberi'),
('S006', 'Temperatura acqua'),
('S006', 'pH acqua'),
('S006', 'Ossigeno disciolto'),
('S007', 'Nitrogeno ammoniacale'),
('S007', 'Densità gamberi'),
('S008', 'Temperatura acqua'),
('S008', 'pH acqua'),
('S009', 'Ossigeno disciolto'),
('S009', 'Nitrogeno ammoniacale');
```

```
CREATE OR REPLACE FUNCTION is_data_successiva(
    p_gamberaia INTEGER,
```

```
    p_parametro VARCHAR(30),
    p_data_ora TIMESTAMP
)
RETURNS BOOLEAN AS $$
BEGIN
    RETURN p_data_ora > COALESCE((SELECT MAX(data_ora) FROM misurazione WHERE gamberaia
= p_gamberaia AND parametro = p_parametro), 'epoch'::timestamp);
END;
$$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE FUNCTION check_misurazione_valida(
    p_gamberaia INTEGER,
    p_parametro VARCHAR(30),
    p_data_ora TIMESTAMP
)
RETURNS BOOLEAN AS $$
DECLARE
    v_sensore_applicato BOOLEAN;
BEGIN
    SELECT COUNT(*) > 0
    INTO v_sensore_applicato
    FROM sensore s
    JOIN caratteristiche_parametro cp ON s.codice_sensore = cp.sensore
    WHERE cp.parametro = p_parametro
    AND s.gamberaia = p_gamberaia
    AND s.applicato_a_gamberaia = TRUE;

    RETURN v_sensore_applicato;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TABLE misurazione (
    data_ora TIMESTAMP NOT NULL,
    gamberaia INTEGER NOT NULL,
    parametro VARCHAR(30) NOT NULL,
    operatore VARCHAR(30),
    isAllarme BOOLEAN NOT NULL DEFAULT FALSE,
    IDAllarme INTEGER CHECK (IDAllarme < 9999 AND IDAllarme > 0),
    valore DECIMAL(8,3) NOT NULL,
    PRIMARY KEY (data_ora, gamberaia, parametro),
    FOREIGN KEY (gamberaia) REFERENCES gamberaia(codice),
    FOREIGN KEY (parametro) REFERENCES parametro(descrizione_parametro),

    CONSTRAINT check_data_successiva CHECK (
        is_data_successiva(gamberaia, parametro, data_ora)
```

```

),
CONSTRAINT check_misurazione_valida CHECK (
    (isAllarme = FALSE AND operatore IS NULL AND IDAllarme IS NULL) OR
    (isAllarme = TRUE AND operatore IS NOT NULL AND IDAllarme IS NOT NULL AND
check_misurazione_valida(gamberaia, parametro, data_ora))
)
);
INSERT INTO misurazione (data_ora, gamberaia, parametro, valore)
VALUES
('2022-01-01 10:00:00', 1, 'Temperatura acqua', 24.5),
('2022-01-01 10:15:00', 1, 'pH acqua', 7.2),
('2022-01-01 11:00:00', 2, 'Ossigeno disciolto', 4.8),
('2022-01-01 11:30:00', 2, 'Nitrogeno ammoniacale', 0.4),
('2022-01-01 12:00:00', 3, 'Densità gamberi', 480.0),
('2022-01-01 12:30:00', 3, 'Temperatura acqua', 25.5);

INSERT INTO misurazione (data_ora, gamberaia, parametro, valore)
VALUES
('2022-01-01 13:00:00', 6, 'Temperatura acqua', 26.0);

```

Query SQL

Workpackage	Task	Responsabile
WP3	SQL: Query	Allocca Francesco Vittorio

5.1. Query con operatore di aggregazione e join: Media della quantità di gamberi prelevati per ogni specie e numero di prelievi effettuati

DESCRIZIONE: la query calcola la media della quantità di gamberi prelevati per ogni specie e il numero di prelievi effettuati. Vengono utilizzate le tabelle Gambero, Ospitalità e Prelievo. La join tra queste tabelle avviene tramite il campo nome_scientifico nella tabella Gambero e il campo Gambero nella tabella Ospitalità, nonché tramite il campo Gamberaia nella tabella Ospitalità e il campo codice_gamberaia nella tabella Prelievo. Il risultato è un elenco di ogni specie di gambero con la media della quantità prelevata e il numero di prelievi effettuati.

```

SELECT g.nome_scientifico, AVG(o.quantità) AS media_quantità, COUNT(p.data) AS num_prelievi
FROM Gambero g
JOIN Ospitalità o ON g.nome_scientifico = o.Gambero
JOIN Prelievo p ON o.Gamberaia = p.codice_gamberaia
GROUP BY g.nome_scientifico;

```

5.2. Query nidificata complessa: Trova tutti i trattamenti applicati a una specifica gamberaia in un anno, con i relativi materiali utilizzati e le quantità

DESCRIZIONE: Questa query utilizza una subquery nidificata per trovare tutti i trattamenti applicati a una specifica gamberaia (in questo caso la gamberaia con codice 1) in un anno, con i relativi materiali utilizzati e le quantità. La subquery interna recupera i codici dei trattamenti applicati alla gamberaia, mentre la query esterna unisce le tabelle Trattamento e Prodotto per ottenere i dettagli dei materiali utilizzati e le quantità.

```
SELECT t.nome_attivita, p.materiale_utilizzato, p.quantita
FROM Trattamento t
JOIN Prodotto p ON t.codice = p.trattamento
WHERE t.codice IN (
    SELECT trattamento
    FROM Applicazione a
    WHERE a.gamberaia = 1 AND EXTRACT(MONTH FROM a.data) BETWEEN 1 AND 12
)
ORDER BY t.nome_attivita;
```

5.3. Query insiemistica: Sensori applicati ad una particolare Gamberaia

DESCRIZIONE: Questa query visualizza per una particolare Gamberaia, i sensori che misurano un determinato parametro. In questo caso Gamberaia 1 e Sensore con parametro "Temperatura Acqua" Viene utilizzato l'operatore INTERSECT per visualizzare solo i sensori applicati a Gamberaia con determinato parametro.

```
SCRIPT:
SELECT s.codice_sensore, s.gamberaia, s.azienda, s.modello, s.descrizione, s.marca
FROM sensore s
INTERSECT
SELECT cp.sensore, s.gamberaia, s.azienda, s.modello, s.descrizione, s.marca
FROM caratteristiche_parametro cp
JOIN sensore s ON cp.sensore = s.codice_sensore
WHERE cp.parametro = 'Temperatura acqua' AND s.gamberaia = 1;
Viene utilizzato l'operatore INTERSECT per visualizzare solo i sensori applicati a Gamberaia con determinato parametro.
```

6. Viste

Workpackage	Task	Responsabile
WP4	Viste	Foschillo Gennaro

6.1. Vista Vista Gamberaie, Gamberi e Trattamenti Associati

DESCRIZIONE: La vista Gamberaie_Gamberi_Trattamenti restituisce una lista di informazioni relative alle gamberaie, ai gamberi ospitati e ai trattamenti applicati.

```
CREATE OR REPLACE VIEW Gamberaie_Gamberi_Trattamenti AS
SELECT g.codice AS gamberaia_codice, g.capienza, g.longitudine, g.latitudine,
       gb.nome_scientifico, gb.categoria, gb.nome_comune, gb.varietà, gb.stagione_allevamento,
       t.codice AS trattamento_codice, t.nome_attivita, t.operatore
FROM Gamberaia g
LEFT JOIN Ospitalità o ON g.codice = o.Gamberaia
LEFT JOIN Gambero gb ON o.Gambero = gb.nome_scientifico
LEFT JOIN Applicazione a ON g.codice = a.gamberaia
LEFT JOIN Trattamento t ON a.trattamento = t.codice;
```

6.1.1. Query con Vista: Elenco delle gamberaie con relativi gamberi e trattamenti

```
SELECT *
FROM Gamberaie_Gamberi_Trattamenti
WHERE stagione_allevamento = 'estate'
AND trattamento_codice IS NOT NULL
ORDER BY gamberaia_codice, nome_scientifico;
```

7. Trigger

7.1. Trigger inizializzazione

Workpackage	Task	Responsabile
WP1	Trigger inizializzazione/popoloamento database	Allocca Francesco Vittorio

7.1.1. Trigger Generazione codice univoco per Gamberaia

DESCRIZIONE: questo trigger genera un codice univoco per ogni nuova inserzione nella tabella Gamberaia, incrementando il valore massimo esistente di 1.

```
CREATE OR REPLACE FUNCTION genera_codice()
RETURNS TRIGGER AS $$
BEGIN
```

```

IF NEW.codice IS NULL THEN
    SELECT COALESCE(MAX(codice)::integer, 0) + 1 INTO NEW.codice
    FROM Gamberaia;
END IF;
RETURN NEW;
END;
$$ LANGUAGE plpgsql;

```

```

CREATE TRIGGER trg_genera_codice
BEFORE INSERT ON Gamberaia
FOR EACH ROW
EXECUTE PROCEDURE genera_codice();

```

7.1.2. Trigger per valori di default sensore

DESCRIZIONE: il trigger si attiva prima di ogni inserimento o aggiornamento di un sensore nella tabella "sensore". Se il campo "gamberaia" del sensore è nullo, imposta i campi "applicato_a_gamberaia", "consumo_energia", "indirizzo_rete" e "posizione" con valori di default (rispettivamente FALSE, NULL, NULL e NULL)

```

CREATE OR REPLACE FUNCTION setta_valori_default_trigger()
RETURNS TRIGGER AS $$
BEGIN
    IF NEW.gamberaia IS NULL THEN
        NEW.applicato_a_gamberaia := FALSE;
        NEW.consumo_energia := NULL;
        NEW.indirizzo_rete := NULL;
        NEW.posizione := NULL;
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

```

```

CREATE TRIGGER setta_valori_default
BEFORE INSERT OR UPDATE ON sensore
FOR EACH ROW
EXECUTE FUNCTION setta_valori_default_trigger();

```

7.2. Trigger per vincoli aziendali

<i>Workpackage</i>	<i>Task</i>	<i>Responsabile</i>
WP4	Trigger per vincoli aziendali	Foschillo Gennaro

La sezione deve contenere almeno due trigger. È possibile aggiungerne altri se necessario

7.2.1. Trigger1: Aggiornamento dell'ospitalità

DESCRIZIONE: questo trigger si attiva dopo l'inserimento di un nuovo record nella tabella PRELIEVO e aggiorna lo stato dell'ospitalità nella tabella OSPITALITA impostando inCorso a FALSE e data_fine con la data dell'ultimo prelievo, solo se l'idoneità dei gamberi è stata verificata.

```
CREATE OR REPLACE FUNCTION update_ospitalita()
RETURNS TRIGGER AS $$
BEGIN
    IF NEW.idoneità_gamberi THEN
        UPDATE ospitalita
        SET inCorso = FALSE, data_fine = NEW.data
        WHERE Gamberaia = NEW.codice_gamberaia AND inCorso = TRUE AND data_inizio = (
            SELECT MAX(data_inizio)
            FROM ospitalita
            WHERE Gamberaia = NEW.codice_gamberaia AND inCorso = TRUE
        );
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER update_ospitalita_trigger
AFTER INSERT ON PRELIEVO
FOR EACH ROW
EXECUTE FUNCTION update_ospitalita();
```

7.2.2. Trigger per misurazione non valida

DESCRIZIONE: Questo trigger si attiva prima di ogni inserimento di una nuova misurazione nella tabella "misurazione". Verifica se il valore della misurazione è maggiore o uguale al valore soglia associato al parametro corrispondente: Se verificato, setta un valore di IDallarme incrementando l'ultimo allarme definito, sceglie un operatore che gestirà l'allarme e imposta a TRUE il valore di isAllarme.

```
CREATE OR REPLACE FUNCTION trg_set_allarme()
RETURNS TRIGGER AS $$
DECLARE
    max_id_allarme INTEGER;
BEGIN
    IF NEW.valore >= (SELECT valore_soglia FROM parametro WHERE descrizione_parametro =
NEW.parametro) THEN
        NEW.isAllarme := TRUE;
        NEW.operatore := (SELECT nome FROM operatore ORDER BY RANDOM() LIMIT 1);
        SELECT COALESCE(MAX(IDallarme), 0) + 1 INTO max_id_allarme FROM misurazione;
        NEW.IDallarme := max_id_allarme;
    ELSE
```

```
        NEW.operatore := NULL;
        NEW.IDAllarme := NULL;
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
CREATE TRIGGER set_allarme
BEFORE INSERT ON misurazione
FOR EACH ROW
EXECUTE FUNCTION trg_set_allarme();
```

7.2.3. Trigger aggiornamento quantità totale

DESCRIZIONE: il trigger si attiva prima di ogni inserimento di un nuovo prelievo nella tabella "prelievo". Aggiorna la quantità totale della giornata di prelievo corrispondente, sommando la quantità del nuovo prelievo alla quantità totale già presente.

```
CREATE OR REPLACE FUNCTION update_quantita_totale()
RETURNS TRIGGER AS $$

    BEGIN
        UPDATE Giornata_di_Prelievo
        SET quantità_totale = quantità_totale + (SELECT quantità FROM ospitalita WHERE
        (Gamberaia = NEW.codice_gamberaia AND data_inizio <= NEW.data AND inCorso =
        TRUE))
        WHERE data = NEW.data;
        RETURN NEW;
    END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER update_quantita_totale_trigger_after
BEFORE INSERT ON PRELIEVO
FOR EACH ROW
EXECUTE FUNCTION update_quantita_totale();
```