

PROGETTO BASI DI DATI

Progettazione ed implementazione di una base di dati relazionale per la gestione delle assistenze di una ditta di gestione di impianti di riscaldamento.

Parata Loris 144338

Brugnera Matteo 137370

Rasera Giovanni 143395

22 aprile 2021



**UNIVERSITÀ
DEGLI STUDI
DI UDINE**

Contatti:

- *144338@spes.uniud.it*
- *137370@spes.uniud.it*
- *143395@spes.uniud.it*

Link utili:

- Github: <https://github.com/GiovanniRaseraF/ProgettoBasiDiDatiLaboratorio>

Indice

1	Introduzione	4
1.1	Introduzione	4
2	Raccolta e analisi dei requisiti	5
2.1	Tabella dei requisiti	5
2.2	Glossario dei termini	5
2.3	Riscrittura dei requisiti	6
2.4	Caso di studio	7
2.5	Requisiti operazionali	7
3	Progettazione concettuale	9
3.1	Definizione specifiche	9
3.2	Schema Entità-Relazione	9
3.2.1	Evidenziazione dello schema E-R	9
3.3	Regole di derivazione	11
3.4	Vincoli d'integrità	11
3.5	Pattern di progettazione	12
4	Progettazione logica	13
4.1	Tabella dei volumi	13
4.2	Tabella delle operazioni	14
4.3	Carico applicativo	15
4.4	Analisi delle ridondanze	15
4.4.1	Tabella dei volumi	15
4.4.2	Tabella delle operazioni	15
4.4.3	Tabella degli accessi in riferimento all'operazione 3	15
4.4.4	Tabella degli accessi in riferimento all'operazione 6	16
4.5	Rimozione delle generalizzazioni	17
4.6	Regole di derivazione	17
4.7	Vincoli d'integrità	17
4.8	Vincoli d'integrità aggiuntivi dello schema E-R ristrutturato	18
4.9	Modello relazionale	18
4.10	Normalizzazione	19
4.10.1	Prima forma normale(1FN)	19
4.10.2	Seconda forma normale(2FN)	19
4.10.3	Terza forma normale(3FN)	19
5	Progettazione fisica	20
5.1	Data Definition Language (DDL)	20
5.1.1	Collegamento	20
5.1.2	Creazione	20
5.1.3	Definizione di tipi personalizzati	20
5.1.4	Cliente	20
5.1.5	Capace Di Risolvere	21
5.1.6	Intervento	21
5.1.7	Tipologia Guasto	21
5.1.8	Richiesta D'Assistenza	22
5.2	Indici	23
5.2.1	Indici implementati	23
5.3	UDF	24
5.3.1	Controllo Inserimento Cliente	24
5.4	Operazioni	25
5.4.1	Operazione 1: inserisci cliente	25
5.4.2	Operazione 2: inserimento richiesta d'assistenza	25
5.4.3	Operazione 3:	25
5.4.4	Operazione 4:	25
5.4.5	Operazione 5:	25
5.4.6	Operazione 6:	25
5.4.7	Operazione 7:	25
5.4.8	Operazione 8:	25

5.4.9	Operazione 9:	25
5.5	Trigger	25
5.5.1	Evento 1	25
5.6	Pulizia	25
5.6.1	Implementazione	25
6	Implementazione	26
6.1	Hardware e Tecnologie Utilizzate	26
6.2	Popolazione della base di dati	26
6.3	Connessione alla base di dati	26
6.4	Preparazione iniziale	26
6.5	Cliente	26
6.6	Disconnessione dalla base dei dati	26
7	Analisi dei dati	27
8	Bibliografia	28

1 Introduzione

1.1 Introduzione

Lo scopo del progetto è quello di implementare un **sistema informativo**, nello specifico una base di dati relazionale, in grado di **gestire le prestazioni di un centro di assistenza per impianti di riscaldamento**.

Le funzionalità che dovrà prevedere il sistema sono di seguito specificate.

L'azienda deve permettere di gestire le **richieste di assistenza**, che sono a loro volta **composte da un insieme di interventi effettuati da tecnici specializzati**.

I servizi offerti sono usufruibili solo da **clienti facenti parte di due tipologie di persone giuridiche**, che rappresentano **aziende** ed **enti pubblici**, oppure da persone fisiche che rappresentano i **singoli cittadini**. Ognuno di essi avrà un **codice identificativo** generato dal sistema una volta divenuti clienti, grazie al quale si potrà risalire a tutti i loro dati. In particolare, per le aziende e gli enti pubblici si vuole tenere il codice di Partita IVA, mentre per i singoli cittadini il Codice Fiscale.

Le assistenze vengono identificate univocamente dal **codice di Richiesta d'Assistenza**. Ogni intervento è legato all'assistenza ed identificato univocamente da un **numero progressivo**. Le richieste di assistenza vengono accettate solamente se la tipologia del problema è presente nella lista di problematiche risolvibili.

Per ogni intervento, si tiene traccia di:

- **cliente** richiedente l'assistenza;
- **tipologia** del **guasto**;
- **seriale** del sistema di riferimento;
- **tecnico** assegnato;
- **codice identificativo dell'intervento**;
- **modalità d'intervento**;
- **data** intervento;
- **durata** intervento.

2 Raccolta e analisi dei requisiti

2.1 Tabella dei requisiti

Questa fase rappresenta l'inizio della realizzazione di un sistema informativo. Con essa si cerca di comprendere quali sono gli obiettivi che vengono richiesti. È necessario porre particolare attenzione alla terminologia utilizzata, in modo tale da poter procedere alla formulazione dei requisiti.

2.2 Glossario dei termini

Il linguaggio naturale è ambiguo, ecco perché è necessario chiarire con precisione ogni termine utilizzato durante questa fase di progettazione. Per ogni termine introdotto si definiscono:

- **Descrizione:** definizione semantica del termine
- **Sinonimi:** eventuali sinonimi utilizzati per identificare lo stesso oggetto
- **Correlazioni:** le relazioni esistenti tra i diversi oggetti

Il seguente glossario definisce i termini più rilevanti che saranno l'input della fase di progettazione concettuale.

Glossario			
Termine	Descrizione	Sinonimo	Correlazione
Cliente	soggetto che effettua una richiesta di assistenza all'azienda di assistenza	Persona Giuridica Persona fisica	Assistenza
Persona fisica	soggetto che rappresenta se stesso e che effettuerà una richiesta di assistenza	Singolo cittadino Cliente	Cliente
Persona giuridica	persona fisica che rappresenta un'azienda o un ente pubblico che effettuerà una richiesta di assistenza	Azienda Ente Pubblico Cliente	Cliente
Richiesta d'assistenza	inizializzazione di un nuovo contratto di assistenza tra un cliente e l'azienda di assistenza, ogni richiesta d'assistenza composta è da almeno un intervento	Non ha sinonimi	Cliente Intervento Tip. di guasto
Intervento	prestazione eseguita da un tecnico per risolvere un guasto	Prestazione	Tip. di guasto Tecnico
Tecnico	dipendente dell'azienda di assistenza, specializzato nella risoluzione di varie tipologie di guasto	Dipendente	Tip. guasto Intervento
Tipologia di guasto	problematica specifica da risolvere	Problema	Tecnico Cliente Assistenza

2.3 Riscrittura dei requisiti

In questa fase vengono tradotte le richieste della consegna in requisiti da soddisfare e viene definito il ruolo delle entità all'interno della base di dati.

Riscrittura dei requisiti
Frasi di natura generale
<ul style="list-style-type: none">- Si vuole implementare un sistema automatico per la gestione dei servizi di assistenza e dei relativi interventi di un'azienda d'assistenza di impianti di riscaldamento- Le assistenze possono essere richieste solamente per problemi risolvibili dai tecnici dell'azienda.
Frasi relative al cliente
<ul style="list-style-type: none">- I clienti possono essere di diversa natura:- Persona Giuridica, che a sua volta può essere un'azienda oppure un ente pubblico;- Singolo cittadino, ovvero persona fisica;- I clienti richiedono un'assistenza relativa ad una specifica tipologia di guasto, specificandone il seriale del sistema di riferimento .- I clienti sono identificati univocamente dal proprio codice fiscale o partita Iva, in base alla tipologia del cliente e dai relativi dati anagrafici del cliente stesso o del referente aziendale, un recapito telefonico, un indirizzo e la città in cui risiede.
Frasi relative alla richiesta d'assistenza
<ul style="list-style-type: none">- Un'assistenza è caratterizzata da un codice identificativo generato dal sistema. Nel momento in cui viene completata l'assistenza viene memorizzata la data dell'ultimo intervento.
Frasi relative all'intervento
<ul style="list-style-type: none">- Un intervento è caratterizzato da :- un numero progressivo, che fa riferimento alla richiesta d'assistenza a cui appartiene:- modalità con la quale può essere eseguito (da remoto o in sede)- dalla data e dalla relativa durata del singolo intervento misurato in ore (0-24).- Un solo intervento può essere eseguito durante l'arco della giornata lavorativa.
Frasi relative al tecnico
<ul style="list-style-type: none">- Un tecnico è identificato dal suo codice fiscale, dai suoi dati anagrafici, un recapito telefonico, una data di assunzione e indirizzo del domicilio.-Inoltre sarà presente il conteggio mensile delle ore lavorate.
Frasi relative al guasto
<ul style="list-style-type: none">- Una tipologia di guasto è rappresentata da:- codice univoco,- descrizione del problema.
Frasi relative al coinvolgimento di cliente, assistenza e tipologia guasto
<ul style="list-style-type: none">- Quando un cliente richiede un'assistenza relativa ad uno specifico guasto, per poter finalizzare e registrare la data di inizio assistenza è necessario che sia presente quella tipologia di guasto tra i guasti riparabili e l'inserimento del numero seriale del sistema di riferimento.
Frasi relative al coinvolgimento di tecnico e tipologia guasto
<ul style="list-style-type: none">- Un tecnico può essere assunto se è capace di risolvere almeno una tipologia di guasto , ma nel tempo può ampliare le proprie competenze e risolvere problemi anche di tipologia diversa.
Frasi relative al coinvolgimento di assistenza, intervento, tecnico e tipologia guasto
<ul style="list-style-type: none">- Una volta effettuata la richiesta di assistenza, si pianifica il primo intervento assegnando un tecnico competente per risolvere quella tipologia di guasto, controllando che in quella data non ci sia un intervento programmato per lo stesso tecnico altrove.

2.4 Caso di studio

Per la realizzazione del sistema informativo è stato *teorizzato un business-plan* di un'azienda di piccole/medie dimensioni che opera a livello regionale. Consideriamo la seguente situazione dopo 10 anni di attività:

- numero **clienti totale**: 10.000 (fedeli e non) ¹;
- numero **tecnici totali**: 75;
- media **clienti annuali**: 10.00;
- **media di 2 richieste di assistenza** per cliente all'anno;
- **media di 3 interventi per ogni assistenza** richiesta;
- in **media 300 clienti rimangono fedeli** al nostro servizio;

Supponiamo che dei 10.000 clienti avuti in 10 anni, 300 siano rimasti fedeli a noi. Di conseguenza l'undicesimo anno operativo dell'azienda otterrà $3.000+1.000=4.000$ clienti (fedeli + nuovi clienti annuali, fedeli e non). Facendo così, avremo $4.000*2=8.000$ assistenze totali e quindi 24.000 interventi. Quest'ultimi vengono divisi tra i 75 tecnici, facendo sì che ognuno di essi abbia in media 320 interventi da eseguire in un anno.

Nel dodicesimo anno otteniamo un totale di 3.300 clienti fedeli, arrivando a fine anno con 4.300 clienti e 25.800 interventi fatti, ovvero 344 a tecnico. Tutto ciò risulta essere troppo oppressivo per un lavoratore, quindi dobbiamo aumentare il numero di tecnici da assumere a fronte dell'aumento della clientela, aumentando così il volume dell'azienda stessa.

Se supponiamo che ogni anno vengono assunti e/o formati 6 nuovi dipendenti, abbiamo che il rapporto interventi/tecnici rimane sempre intorno ai 320 (ovvero ogni tecnico avrà 320 interventi all'anno). Ad esempio il tredicesimo anno avremmo 87 operai con 27.600 interventi, cioè intorno ai 317 interventi per operaio.

Ultimo appunto importante riguarda la distinzione che è possibile fare tra i vari clienti. Più in particolare del 100% dei clienti, il 50% risulta essere un ente pubblico mentre il restante 50% è privato. Del 50% pubblico, il suo 70% fa riferimento alle aziende, mentre il 30% agli enti pubblici.

2.5 Requisiti operazionali

Le **operazioni principali** che prenderemo in considerazione sono:

- **operazione 1**: date le informazioni di un nuovo cliente, inserirlo nella base di dati → *19 volte a settimana*;
- **operazione 2**: date le informazioni necessarie fornite da un cliente, aggiungere una richiesta d'assistenza relativa ad una tipologia di problema → *150 volte a settimana*;
- **operazione 3**: data l'informazione di una richiesta d'assistenza, creare un nuovo intervento relativo ad essa → *450 volte a settimana*;
 - **operazione 3.1**: Ricerca e classificazione in base al numero di interventi eseguiti nel mese corrente dei tecnici capaci di risolvere la tipologia di guasto presente nella richiesta di assistenza.
 - **operazione 3.2**: Assegnazione del tecnico all'intervento.
 - **operazione 3.3**: Creazione dell'intervento.
- **operazione 4**: date le informazioni di un nuovo dipendente e delle tipologie di guasti che è capace di risolvere, aggiungerlo alla base di dati → *6 all'anno*;
- **operazione 5**: dato il codice identificativo del cliente, visualizzare le relative richieste d'assistenza effettuate → *30 a settimana*;
- **operazione 6**: visualizzare il numero di ore mensili di lavoro effettuate da ogni tecnico → *26 al mese*;
- **operazione 7**: dato la tipologia di guasto, il numero di interventi eseguiti su di esso, ed i tecnici specializzati su quella tipologia, visualizzare per ogni tipologia guasto la relativa percentuale di forza lavoro disponibile a risolverla. → *1 volta all'anno*.
Questo dato può essere utilizzato per vedere in quali reparti ci sono carenze di tecnici, andando così a mirare l'assunzione/training dei nuovi dipendenti;

¹per fedeli si intende tutti quei clienti che faranno riferimento a noi come aziende per qualsiasi guasto negli anni a venire

- **operazione 8:** dato il codice identificativo di una richiesta di assistenza, visualizzare lo storico degli interventi → *10 volte al giorno*.
Si vuole monitorare il più possibile l'impiego dei tecnici, cioè chi è impegnato e dove;
- **operazione 9:** dato il codice identificativo di un cliente ed una data, visualizzare il tempo complessivo degli interventi effettuati da quel momento → *1 volta a settimana*;

3 Progettazione concettuale

3.1 Definizione specifiche

Per scrivere il nostro schema e per descrivere le specifiche e realizzare la relazione abbiamo utilizzato questa convenzione:

- Entità: prima lettera MAIUSCOLA e il resto minuscolo
Esempio: Assistenza, Cliente, Intervento.
- Relazione: tutta la parola minuscola
Esempio: è capace di risolvere, composta da.
- Attributi delle entità e delle relazioni: prima lettera minuscola e il resto della parola cammellizzato
Esempio: dataFineAssistenza, numeroIntervento, dataAssunzione.
- Rappresentazione di un attributo di un entità o di una relazione: utilizzato il simbolo \rightarrow tra l'entità e l'attributo
Esempio: Entità \rightarrow attributoUno, Entità \rightarrow attributoDue, Intervento \rightarrow modalità.

3.2 Schema Entità-Relazione

Il modello Entità-Relazione (E-R) è il modello teorico utilizzato in questa fase di progettazione concettuale per la rappresentazione grafica e concettuale dello scenario di interesse. Questo permette di comprendere in modo semplice ed intuitivo quali sono i soggetti principali della scena e come sono relazionati tra di loro. Modello E-R dell'azienda di assistenza:

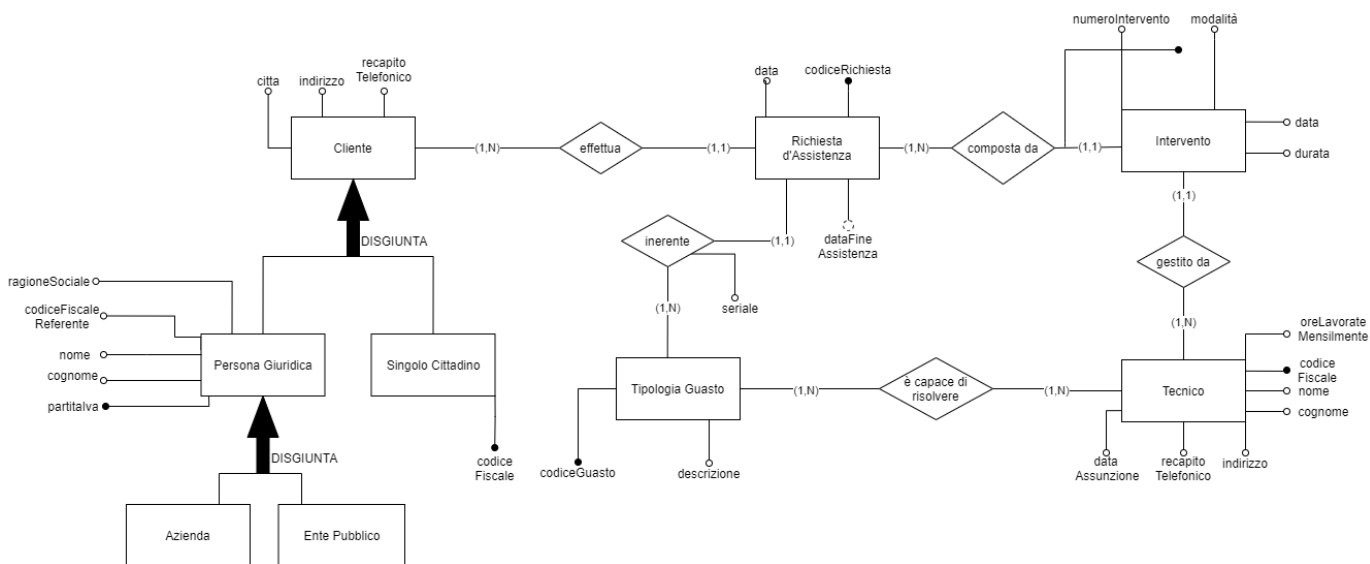


Figura 1: Schema E/R complessivo

3.2.1 Evidenziazione dello schema E-R

Figura 2 in questa porzione di schema ER troviamo un ciclo. Esso è composto dalle seguenti Entità:

- Richiesta d'Assistenza;
- Intervento;
- Tecnico;
- Tipologia Guasto;

Intervento è relazionato a Tecnico tramite "gestito da", una relazione **uno a molti**. Infatti, un Intervento è gestito da un singolo Tecnico in giornata, mentre un Tecnico può gestire uno o più Interventi. Quest'ultimo però può risolvere solo un numero finito di tipologie di problemi, descritti dalla relazione **molti a molti** "è capace di risolvere", dove un Tecnico è in grado di risolvere più problemi, ed essi possono essere risolti da più Tecnici.

Quest'ultimo fa sì che il ciclo non sia problematico perché un Tecnico che non ha le conoscenze per riparare un Guasto non può farlo.

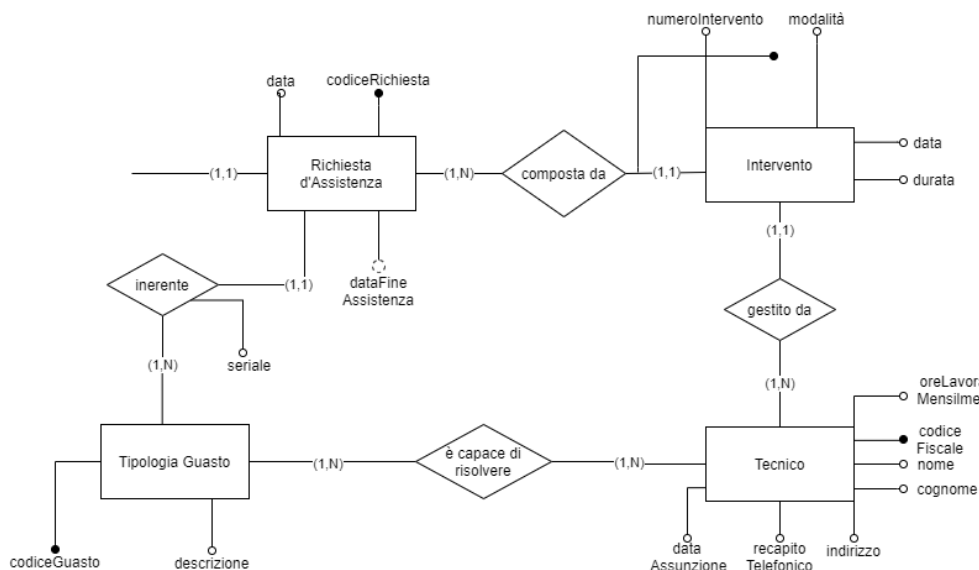


Figura 2: Ciclo

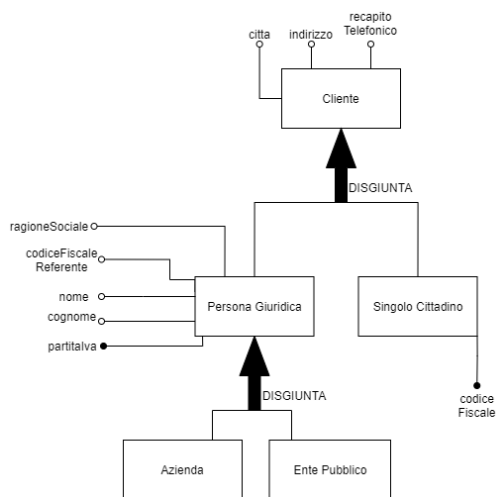


Figura 3: Generalizzazione

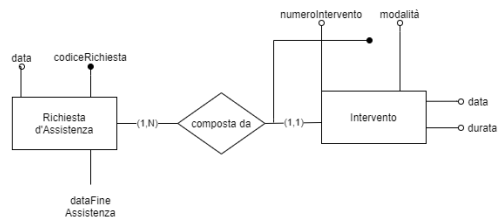


Figura 4: Entità debole

Infine, la Tipologia Guasto è legato con Richiesta d'Assistenza tramite la relazione **uno a molti "inerente"**, dove una Tipologia Guasto può essere inerente a più Richieste d'Assistenza, mentre una Richiesta fa riferimento ad una sola Tipologia di Guasto.

Figura 3 lo schema mette in evidenza la presenza di una generalizzazione esclusiva totale tra le entità Persona Giuridica e Singolo Cittadino che differiscono per codice identificativo univoco. Una Persona Giuridica viene distinta tramite Partita Iva, mentre un Singolo Cittadino dal proprio CodiceFiscale. L'entità Cliente mi rappresenta il sovrainsieme a cui facciamo riferimento. Esso viene poi classificato attraverso l'utilizzo di due specializzazioni, individuando così i suoi sottoinsiemi, ovvero tutti gli elementi che compongono l'insieme stesso.

Con la prima generalizzazione totale indichiamo che un Cliente deve essere una Persona Giuridica o un Singolo Cittadino (quindi persona fisica). Questa netta distinzione è data dalla totalità (indicata dalla freccia nera), partizionando l'insieme in due sottoinsiemi.

Analogo discorso viene fatto con la Persona Giuridica, specializzandola in Azienda ed Ente Pubblico. Inoltre, entrambe le specializzazioni sono DISGIUNTE perché non può succedere che una persona giuridica sia allo stesso tempo una persona fisica.

Un caso critico che si può presentare a causa della sua forma (gerarchie di specializzazione) riguarda l'ereditarietà multipla. Questo problema però non viene riscontrato in questo tipo di schema in quanto non abbiamo entità che appartengono a più gerarchie di specializzazione.

Infine, è importante notare come tutte le entità ereditino gli attributi da cui derivano, più in particolare tutti gli attributi di Cliente verranno ereditati da qualsiasi entità appartenente a questa porzione di schema e le entità Azienda ed Ente Pubblico vengono individuate univocamente dagli attributi derivanti da Persona Giuridica.

Figura 4 lo schema mette in evidenza la presenza di un'entità debole che necessita della presenza di una richiesta d'assistenza per poter essere distinta univocamente. Prima cosa da notare è la scelta di

un attributo opzionale (dataFineAssistenza) in quanto esso verrà aggiunto una volta terminato l'ultimo intervento. Inoltre, la relazione "composta da" è una relazione **molte a uno** perché una Richiesta di Assistenza può essere composta da molti interventi, mentre un intervento appartiene ad una sola specifica richiesta.

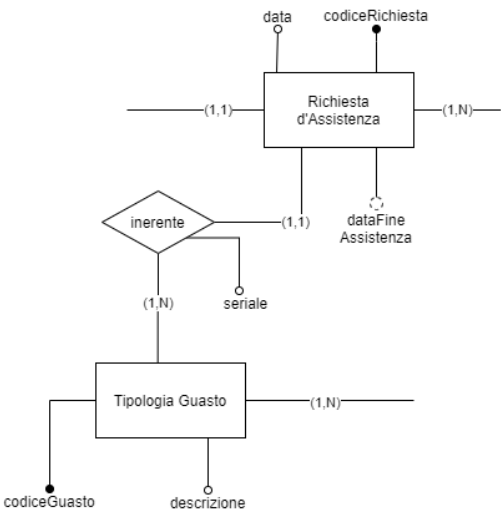


Figura 5: Richiesta d'Assistenza inerente ad una tipologia di guasto

Figura 5 in questa figura si nota la modellazione della relazione tra Richiesta d'Assistenza e Tipologia Guasto:
 E' stato deciso di collegare alla Richiesta d'Assistenza un seriale, cioè il numero di serie del sistema per il quale si vuole essere assistiti; non necessariamente a quest'ultimo deve far riferimento un sistema specifico, siccome ciò a cui siamo interessati è il tipo di guasto presente.
 Sarà compito di un livello di astrazione superiore il controllo dell'autenticità della Richiesta d'Assistenza.

3.3 Regole di derivazione

Le regole di deriavazione sono delle informazioni ottenute tramite concetti dello schema, oppure mediante delle deduzioni. Nel modello concettuale analizzato è presente un attributo derivato, che permette ad ogni Tecnico di salvare il numero di ore lavorate mensilmente.
 Questo permette di non dover scansionare tutti gli Interventi presenti nella base di dati per ricavare l'informazione. L'attributo **Tecnico->oreLavorateMensilmente** sarà quindi aggiornato ogni volta che un Tecnico verrà assegnato ad un Intervento.
 E' possibile calcolarlo sommando tutti i valori di **Intervento->durata** che si riferiscono al Tecnico in questione e al mese corrente.
 Ogni volta che il mese finisce deve essere azzerato.

3.4 Vincoli d'integrità

I vincoli d'integrità sono tutte le proprietà di una base-di-dati, sono esprimibili tramite dei predicati; questi devono essere veri per garantire la validità dello schema.

Di seguito sono riportati i vincoli d'integrità presenti nel modello E-R:

- L'attributo **Richiesta d'Assistenza->dataFineAssistenza** deve essere maggiore o uguale alla proprietà **Richiesta d'Assistenza->data**
- Se **Richiesta d'Assistenza->dataFineAssistenza** non è **NULL** allora: il numero di Interventi appartenenti ad un'Assistenza deve essere minore o uguale alla differenza (in giorni) tra **Richiesta d'Assistenza->data** e **Richiesta d'Assistenza->dataFineAssistenza**.
 Questo vincolo deriva dal fatto che è possibile fare un Intervento al giorno per singola Richiesta d'Assistenza.
- L'attributo **Intervento->durata** deve essere un valore nel dominio (0, 24). Il valore della durata è rappresentato tramite il sistema orario a 24 ore.

- Un Tecnico non deve essere assegnato a degli Interventi diversi nello stesso giorno.
- Un Tecnico non deve essere assegnato a degli Interventi facenti parte di una Richiesta d'Assistenza, relativa ad una determinata Tipologia di Guasto, da lui non risolvibili.
- L'attributo **Tecnico->oreLavorateMensilmente** deve essere uguale alla somma di tutti i valori di **Intervento->durata** a cui un Tecnico ha lavorato nel mese corrente.

3.5 Pattern di progettazione

I pattern di progetto sono delle soluzioni progettuali a problemi comuni. Si elencano di seguito i pattern più utilizzati nella progettazione concettuale in basi di dati, applicati allo schema concettuale illustrato precedentemente:

- Relazione "parte-di":
 - Un'assistenza è composta da diversi interventi

4 Progettazione logica

Lo scopo della progettazione logica è di costruire uno schema relazionale che rappresenti in modo accurato, efficiente e soprattutto correttamente tutte le informazioni descritte da uno schema ER prodotto durante la fase precedente.

Questo non è una semplice trasformazione da un modello ad un altro per due motivi:

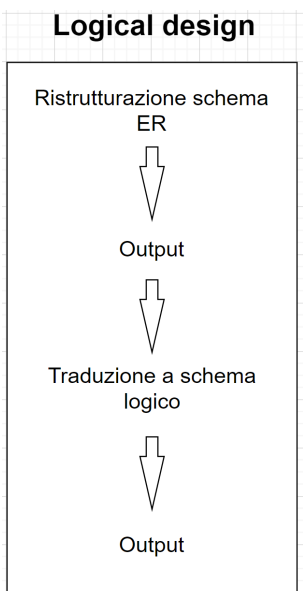
- non tutti i costrutti del modello ER possono essere tradotti nel modello relazionale;
- lo schema deve essere ristrutturato in modo che l'esecuzione delle operazioni avvenga il più efficientemente possibile

Inoltre si controllano e governano le ridondanze. Infatti per analizzarle si usano:

- i volumi dei dati;
- operazioni attese;
- frequenza delle operazioni;

È utile dividere questo tipo di progettazione in due semplici step:

- ristrutturazione dello schema ER, basato sull'ottimizzazione e semplificazione dello schema;
- traduzione nel modello logico.



4.1 Tabella dei volumi

In questa sezione andiamo a definire il numero di occorrenze per ogni entità e relazione presente all'interno dello schema ER. Viene ipotizzato che i volumi facciano riferimento all'attività dopo i suoi dieci anni di vita.

Tabella dei volumi		
Entità/Relazione	Tipo	Volume
Azienda	E	3500
Ente Pubblico	E	1500
Persona Giuridica	E	5000
Singolo Cittadino	E	5000
Cliente	E	10000
effettua	R	200000
Richiesta d'Assistenza	E	200000
inerente	R	200000
Tipologia Guasto	E	30
composto da	R	600000
Intervento	E	600000
gestito da	R	600000
è capace di risolvere	R	750
Tecnico	E	75

4.2 Tabella delle operazioni

Per ogni operazione indicata precedentemente, andiamo a definire la frequenza con la quale essa viene eseguita e la sua tipologia:

- **Batch:** operazioni che si possono "ignorare", ovvero vengono svolte quando il sistema non lavora in pieno regime (ad esempio tarda sera). Facendo così, si lascia spazio alle operazioni più importanti; **Utilizzeremo una " B ".**
- **Interactive:** operazioni più importanti, dove la velocità di esecuzione deve essere veloce. Il tempo di risposta quindi deve essere veloce. **Utilizzeremo una " I ".**

Tabella delle operazioni		
Operazione	Tipo	Frequenza
Operazione 1	I	19 volte a settimana
Operazione 2	I	150 volte a settimana
Operazione 3	I	450 a settimana
Operazione 4	B	6 volte all'anno
Operazione 5	I	30 volte a settimana
Operazione 6	B	1 al giorno
Operazione 7	B	1 volta all'anno
Operazione 8	I	10 volte al giorno
Operazione 9	B	1 volta a settimana

4.3 Carico applicativo

E' necessario calcolare il carico applicativo per scegliere tra le varie tipologie di traduzione quale adottare. Esso viene calcolato ed analizzato mediante la tabella dei volumi e la tabella delle operazioni.

4.4 Analisi delle ridondanze

Nello schema ER è presente una singola ridondanza nell'entità Tecnico. Le ridondanze risultano essere utili perché possono essere informazioni più efficienti da ricavare direttamente invece che utilizzare apposite query per la loro determinazione. Essendo quindi derivabili, bisogna tenere aggiornata questa informazione.
Per tenere una ridondanza presente all'interno dello schema, dobbiamo analizzare tutti i costi relativi ad esso e confrontarli con quelli che si ottengono qualora si ottenesse quella informazione in modo derivato.

4.4.1 Tabella dei volumi

Analizziamo tutte le componenti che si interfacciano con la ridondanza:

Elementi coinvolti		
Tecnico	E	75
gestito da	R	600.000
Intervento	E	600.000

4.4.2 Tabella delle operazioni

Analizziamo tutte le operazioni che agiscono direttamente con la ridondanza:

Operazioni coinvolte		
Operazione 3	I	450 a settimana
Operazione 6	B	1 al giorno

4.4.3 Tabella degli accessi in riferimento all'operazione 3

Il costo di una operazione (calcolato usando una tabella dei volumi) e lo schema di navigazione possono essere riassunti nella tabella degli accessi:

Tabella degli accessi senza la presenza di ridondanza			
Concept	Type	Accesses	Type
Tecnico	E	1	R
gestito da	R	1	R
Intervento	E	1	W

Analizziamo ora i costi:

- $75 * 2 = 150$ (costi in riferimento all’accesso a Tecnico e gestito da)
- $75 * 2 = 150$ (costi in riferimento alla scrittura in Intervento)
- $150 + 150 = 300$ (costo totale giornaliero dell’operazione)

Tabella degli accessi con la presenza di ridondanza			
Concept	Type	Accesses	Type
Tecnico	E	1	W
gestito da	R	1	R
Intervento	E	1	W

Analizziamo ora i costi:

- $75 * 2 = 150$ (costi in riferimento ala modifica di Tecnico)
- 75 (costi in riferimento alla relazione gestito da)
- $75 * 2 = 150$ (costi in riferimento alla scrittura in Intervento)
- $150 + 75 + 150 = 375$ (costo totale giornaliero dell’operazione)

Come possiamo notare, il costo con la presenza di ridondanza è leggermente maggiore (esattamente del 25%), ma come vedremo nella prossima operazione, questo ci comporterà un beneficio.

4.4.4 Tabella degli accessi in riferimento all’operazione 6

Il costo di una operazione (calcolato usando una tabella dei volumi) e lo schema di navigazione possono essere riassunti nella tabella degli accessi:

Tabella degli accessi in presenza di ridondanza			
Concept	Type	Accesses	Type
Tecnico	E	1	R

Tenendo in considerazione che il costo di una operazione di read è uguale ad un accesso, con la presenza della ridondanza, ottenere l’informazione derivabile (Tecnico->oreLavorateMensilmente) ci viene a costare esattamente 75 accessi al giorno cioè:

- $75 * 1 = 75$ accessi giornalieri
- $75 * 26 = 1950$ accessi totali mensili

Tabella degli accessi senza la presenza di ridondanza			
Concept	Type	Accesses	Type
Tecnico	E	1	R
gestito da	R	1	R
Intervento	E	1	R

Senza la presenza della ridondanza, è necessario:

- $75 * 3 = 225$ accessi al giorno
- $225 * 26 = 5850$ accessi mensili

Conclusioni:

- dall'analisi svolta, vediamo come la presenza della ridondanza risulti conveniente in quanto il numero di accessi è inferiore del 67%
- inoltre, c'è da tenere in considerazione che ogni volta che si effettua l'accesso alla tabella degli Interventi, bisogna filtrare 600.000 righe

4.5 Rimozione delle generalizzazioni

Per tradurre lo schema E-R bisogna rimuovere le varie generalizzazioni:

Accorpamento dei figli della generalizzazione nel genitore

- Le entità Azienda e Ente Pubblico vengono eliminate e le loro proprietà vengono aggiunte a Persona Giuridica. Per distinguerli viene aggiunto l'attributo **tipo**.
Il suo valore è un carattere che assume **A** oppure **E**, in cui il primo indica azienda ed il secondo ente pubblico.
Si effettua questo accorpamento perché le attività svolte sono identiche.
La superchiave **partitaIva** è la medesima, perciò non distinguibili senza la presenza della tipologia.
- Le entità Persona Giuridica e Singolo Cittadino vengono eliminate e le loro proprietà vengono aggiunte a Cliente.
Per distinguerle viene utilizzato come superchiave un unico attributo denominato **codiceFiscale/partitaIva**, con valore rispettante le caratteristiche di un reale numero di Partita Iva, nel caso in cui si tratta di una persona giuridica, e quelle di un reale Codice Fiscale nel caso in cui si tratta di un singolo cittadino.
L'attributo **tipo** di Persona Giuridica diventa opzionale a seconda della tipologia della superchiave.
Si effettua questo accorpamento perché le attività svolte sono identiche.

4.6 Regole di derivazione

Tecnico->oreLavorateMensilmente: attributo derivato che contiene la somma delle ore che un Tecnico ha lavorato nel mese corrente ed è un valore che deve essere maggiore o uguale a zero.

Vale sempre che:

- $0 \leq \text{Tecnico->oreLavorateMensilmente}$
 - sarà zero nel momento in cui il Tecnico non ha ancora lavorato nel mese corrente
- $\text{Somma}(\text{Intervento->durata} \text{ inerente al tecnico interessato}) = \text{Tecnico->oreLavorateMensilmente}$
 - verrà aggiornato ogni volta che viene aggiunto un Intervento a cui un Tecnico lavorerà

4.7 Vincoli d'integrità

1. Nel ciclo tra [Richiesta d'Assistenza, Intervento, Tecnico, Tipologia Guasto]
 - Un Tecnico non deve essere assegnato ad un Intervento che fa riferimento ad una Richiesta d'Assistenza inerente ad una Tipologia Guasto che quel determinato tecnico non sa risolvere
2. Dopo che ho assegnato un valore all'attributo **Richiesta d'Assistenza->dataFineAssistenza** non possono esistere degli Interventi inerenti a quella Richiesta che hanno l'attributo **Intervento->data** maggiore di **Richiesta d'Assistenza->dataFineAssistenza**

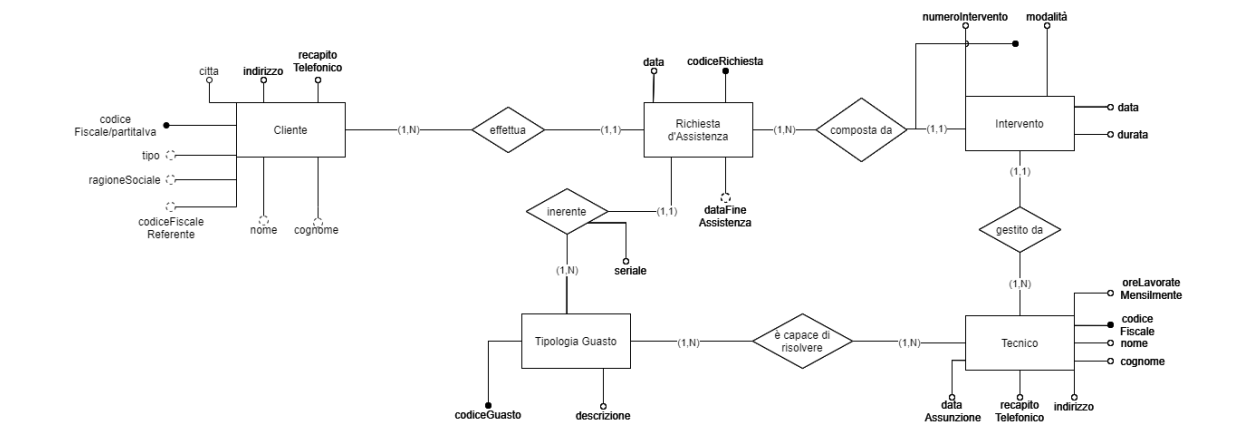


Figura 6:

4.8 Vincoli d'integrità aggiuntivi dello schema E-R ristrutturato

4.9 Modello relazionale

Cliente
<p> Cliente(<u>codiceFiscale/partitaIva</u>,indirizzo,città,recapitoTelefonico,ragioneSociale,codiceFiscaleReferente,nome,cognome, tipo) </p> <p> Unique→codiceFiscaleReferente </p> <p> NotNull→indirizzo,città, recapitoTelefonico </p>
Richiesta d'Assistenza
<p> Richiesta d'Assistenza(<u>codiceRichiesta</u>,data,dataFineAssistenza,<i>codiceCliente*</i>,<i>codiceGuasto*</i>, seriale) </p> <p> NotNull→data, seriale </p> <p> codiceCliente FK→Cliente.partitaIva/codiceFiscale </p> <p> codiceGuasto FK→TipologiaGuasto.codiceGuasto </p>
Intervento
<p> Intervento(<u>codiceRichiesta,numeroIntervento</u>,modalità,data,durata,<i>cfTecnico*</i>) </p> <p> NotNull→modalità,data,durata </p> <p> cfTecnico FK→Tecnico.codiceFiscale </p>

Tecnico
<p>Tecnico(<u>codiceFiscale</u>,nome,cognome,indirizzo,recapitoTelefonico,dataAssunzione,oreLavorateMensilmente)</p> <p>NotNull->nome,cognome,indirizzo,recapitoTelefonico,dataAssunzione,oreLavorateMensilmente</p> <p>cfTecnico FK->Tecnico.codiceFiscale</p>

è capace di risolvere
<p>è capace di risolvere(<u>codiceFiscale</u>,<u>codiceGuasto</u></p>

Tipologia Guasto
<p>Tipologia Guasto(<u>codiceGuasto</u>,descrizione)</p> <p>NotNull->descrizione</p>

4.10 Normalizzazione

4.10.1 Prima forma normale(1FN)

Per vedere se è in Prima Forma Normale, dobbiamo semplicemente controllare che tutti gli attributi dello schema abbiano domini "atomici", ovvero che non siano composti o multivalore. E' questo il caso.

4.10.2 Seconda forma normale(2FN)

Il modello logico relazionale è in Seconda Forma Normale perché ogni attributo non-primο, cioè non appartenente a nessuna chiave, dipende completamente da ogni chiave (ovvero non dipende solamente da una parte di chiave).

4.10.3 Terza forma normale(3FN)

Il modello risulta essere in Terza Forma Normale in quanto, per ogni dipendenza funzionale (non banale) $Y \rightarrow Z$ definita su di esso, Y è una superchiave oppure ogni attributo in Z è primo, cioè contenuto in almeno una chiave del modello.

5 Progettazione fisica

5.1 Data Definition Language (DDL)

Nell'SQL, la definizione dei dati o DDL è una sintassi per creare e modificare oggetti nel database come tabelle, indici e utenti. Esso è legato alla creazione del dizionario dei dati e l'eventuale definizione delle strutture di memorizzazione e dei metodi di accesso ai dati e rappresenta il primo passo della progettazione fisica.

5.1.1 Collegamento

```
-- Bash
sudo -i -u postgres

psql
--
```

Figura 7: Collegamento dal terminale

Il codice in figura va eseguito da un terminale linux con installato postgresql come servizio di sistema.

Per controllare se è presente il servizio postgresql basta digitare: **sudo service postgresql start**, se il comando da un errore sarà necessario installarlo seguendo questa guida: Come installare postgres.

5.1.2 Creazione

```
DROP SCHEMA IF EXISTS assistenzaGuasti CASCADE;

CREATE SCHEMA assistenzaGuasti;

SET search_path TO assistenzaGuasti;
```

Figura 8: Creazione Database

5.1.3 Definizione di tipi personalizzati

5.1.4 Cliente

```
CREATE TABLE cliente(
    codiceFiscalePartitaIva    varchar(16)    NOT NULL,
    indirizzo                  varchar(20)    NOT NULL,
    recapitoTelefonico          varchar(10)    NOT NULL,
    tipo                        character(1)    ,
    codiceFiscaleReferente      varchar(16)    ,
    nome                        varchar(20)    ,
    cognome                     varchar(20)    ,

    CONSTRAINT cliente_pkey
    PRIMARY KEY                (codiceFiscalePartitaIva)
);
```

Figura 9: Definizione tabella Clienti in SQL

Nell'immagine notiamo che alcuni attributi possono essere nulli, questo perchè alcune tipologie di clienti non serve che abbiamo attributi settati ma, possono averli NULL. La query per la creazione del cliente presenta delle differenze rispetto allo schema ER, infatti c'è bisogno di aggiungere l'attributo tipo per distinguere la tipologia di cliente a cui ci stiamo riferendo.

Non basterà solamente aggiungere l'attributo tipo, sarà necessario creare un trigger apposito che, al

momento dell'inserimento, deciderà quali campi lasciare nulli e quelli dovranno obbligatoriamente essere assegnati.
Di seguito viene presentato il trigger per garantire un giusto inserimento del cliente:

```
CREATE TRIGGER triggerInserimentoCliente BEFORE INSERT OR UPDATE
FOR EACH ROW
EXECUTE PROCEDURE controlloInserisciCliente();
```

Figura 10: Definizione tabella Clienti in SQL

Trigger inserimento cliente Il trigger ha bisogno di una UDF per funzionare, in particolare questa funzione dovrà controllare che il vengano settati gli attributi giusti al variare dell'attributo tipo.

5.1.5 Capace Di Risolvere

```
CREATE TABLE capaceDiRisolvere(
  codiceGuasto          integer          NOT NULL,
  cfTecnico              varchar(16)      NOT NULL,

  CONSTRAINT capaceDiRisolvere_pkey
  PRIMARY KEY            (codiceGuasto, cfTecnico),
  CONSTRAINT cfTecnico
  FOREIGN KEY             (cfTecnico)
  REFERENCES              tecnico(codiceFiscale)
  ON UPDATE               CASCADE
  ON DELETE               CASCADE,
  CONSTRAINT codiceGuasto
  FOREIGN KEY             (codiceGuasto)
  REFERENCES              tipologiaGuasto(codiceGuasto)
  ON UPDATE               CASCADE
  ON DELETE               CASCADE
);
```

Figura 11: Definizione tabella Capace Di Risolverei in SQL

Capace di Risolvere presenta una chiave primaria e due chiavi esterne che vengono definite come tali nella parte sottostante alla dichiarazione degli attributi.
E' importante definire le chiavi esterne con le caratteristiche di ON UPDATE CASCADE ON DELETE CASCADE così da tenere sempre aggiornate le chiavi esterne in caso di modifiche alle tabelle puntate e, in caso di eliminazione delle tabelle puntate, la conseguente eliminazione dalla tabella capaceDiRisolvere.

5.1.6 Intervento

fdsa

5.1.7 Tipologia Guasto

Di seguito viene effettuata un'analisi del codice.

```

CREATE TABLE intervento(
    numeroIntervento      integer          NOT NULL,
    data                  timestamp        NOT NULL,
    durata                integer          NOT NULL,
    modalita              character(1)     NOT NULL,
    codiceRichiesta       integer          NOT NULL,
    cfTecnico             varchar(16)     NOT NULL,

    CONSTRAINT interventi_pkey
    PRIMARY KEY           (codiceRichiesta, numeroIntervento),
    CONSTRAINT cfTecnico
    FOREIGN KEY           (cfTecnico)
    REFERENCES            tecnico(codiceFiscale)
    ON UPDATE             CASCADE
    ON DELETE             CASCADE
);

```

Figura 12: Definizione tabella intervento in SQL

```

CREATE TABLE tipologiaGuasto(
    codiceGuasto          integer          NOT NULL,
    descrizione           text             NOT NULL,

    CONSTRAINT tipologiaGuasto_pkey
    PRIMARY KEY           (codiceGuasto),
);
-- Auto Increment
ALTER TABLE tipologieGuasto
ALTER COLUMN codiceGuasto ADD GENERATED ALWAYS AS IDENTITY (
    SEQUENCE      NAME      tipologiaGuasto_codiceGuasto_seq
    START         WITH      1
    INCREMENT     BY        1
    NO
    NO             MINVALUE
    NO             MAXVALUE
    CACHE         1
);

```

Figura 13: Definizione tabella Tipologia Guasto in SQL

5.1.8 Richiesta D'Assistenza

Di seguito viene effettuata un'analisi del codice.

```

CREATE TABLE richiestaDAssistenza(
  codiceRichiesta      integer          NOT NULL,
  data                 timestamp        NOT NULL,
  seriale              varchar(10)     NOT NULL,
  dataFineAssistenza   date            ,
  codiceCliente        varchar(16)     NOT NULL,

  CONSTRAINT richiestaDAssistenza_pkey
    PRIMARY KEY          (codiceRichiesta),
  CONSTRAINT codiceCliente
    FOREIGN KEY          (codiceCliente)
    REFERENCES           cliente(codiceFiscalePartitaIva)
    ON UPDATE            CASCADE,
);
-- Auto Increment
ALTER TABLE richiestaDAssistenza
  ALTER COLUMN codiceRichiesta ADD GENERATED ALWAYS AS IDENTITY (
    SEQUENCE      NAME      richiestaDAssistenza_codiceRichiesta_seq
    START         WITH      1
    INCREMENT     BY        1
    NO
    NO
    CACHE         1
  );

```

Figura 14: Definizione tabella Richiesta D'Assistenza in SQL

5.2 Indici

5.2.1 Indici implementati

5.3 UDF

5.3.1 Controllo Inserimento Cliente

```
CREATE OR REPLACE FUNCTION controlloInserisciCliente()
RETURN TRIGGER LANGUAGE PLPGSQL AS
$$
DECLARE
    tipo          character(1);
BEGIN
    -- Triggere di controllo
    tipo := new.tipo;

    -- Verifico il codice fiscale
    if not verificaCodiceFiscale(tipo)
       or not verificaCodiceFiscale(new.codiceFiscaleReferente) then
        return NULL;

    -- Verifico che sia stata settata
    -- una delle possibili tipologie di cliente
    if tipo = 's' or tipo = 'S' then
        -- Singolo cittadino
        if new.nome = NULL or new.cognome = NULL then
            return NULL;

        return new;
    else
        if tipo = 'a' or tipo = 'A' then
            -- Azienda
            if new.codiceFiscaleReferente = NULL then
                return NULL;

            return new;
        else
            if tipo = 'e' or tipo = 'E' then
                -- Ente pubblico
                if new.codiceFiscaleReferente = NULL then
                    return NULL;

                return new;
            else
                -- Opzione non possibile
                return NULL;
            end if;
        end if;
    end if;
END;
$$
```

Figura 15: Controllo Inserimento cliente

La funzione si occupa di controllare che un cliente sia Singolo Cittadino, Azienda, Entepubblico. Nel caso del Singolo Cittadino devo settare anche nome e cognome, invece nel caso di Azienda o Ente Pubblico devo settare il codiceFiscaleReferente.

5.4 Operazioni

5.4.1 Operazione 1: inserisci cliente

```
INSERT INTO cliente
VALUES(
    $cf_pi,
    $indirizzo,
    $recapito_tel,
    $nome,
    $cognome,
    $tipo,
    $cf_ref
)
```

Figura 16: InserimentoCliente

Per l’inserimento viene usata questa query, al posto del **\$param** verrà messo il vero e proprio valore da inserire

5.4.2 Operazione 2: inserimento richiesta d’assistenza

```
INSERT INTO richiestaAssistenza
VALUES(
    $id,
    $nome,$cognome,
    $data_creazione,
    $data_fine_assistenza,
    $indirizzo,
    $id_cliente_gestito,
    $id_cliente_fornitore
)
```

Figura 17: inserimento richiesta d’assistenza

5.4.3 Operazione 3:

5.4.4 Operazione 4:

5.4.5 Operazione 5:

5.4.6 Operazione 6:

5.4.7 Operazione 7:

5.4.8 Operazione 8:

5.4.9 Operazione 9:

5.5 Trigger

5.5.1 Evento 1

5.6 Pulizia

5.6.1 Implementazione

6 Implementazione

6.1 Hardware e Tecnologie Utilizzate

Per testare e condividere il database a tutti i membri abbiamo creato un server utilizzando un **Raspberry Pi 3** con all'interno **Raspbian**.

- CPU: Arm Cortex-A7
- RAM: 1GB

Dopo aver installato il sistema operativo abbiamo potuto interfacciarci tramite **openssh-server**. A questo punto abbiamo installato **postgresql-11** seguendo le istruzioni riportate dalla guida ufficiale.

- Installazione: <https://www.postgresql.org/download/linux/ubuntu/>

Abbiamo quindi a disposizione la console di postgres che ci permette di eseguire le query simile all'interazione **tty**; abbiamo quindi deciso di permettere a postgres di accettare le connessioni da parte di client **pgAdmin4**.

Per farlo c'è bisogno di modificare i file di configurazioni iniziale di postgresql; nel nostro caso si trovano in

/etc/postgresql/11/main, più precisamente dobbiamo modificare i file:

- **pg_hba.conf**: Nello specifico va inserita un riga che specifica quali sono gli indirizzi IP di cui postgres può fidarsi.
- **postgresql.conf**: Nello specifico va aggiunta la riga **listen_addresses = '*'**.

E' importante sottolineare che: !QUESTE IMPOSTAZIONI DEL SERVER SERVONO A FAR SI CHE L'ACCESSO E IL TESTING SIA PIU' VOLCE, NON E' PER NULLA SICURO USARE QUESTE IMPOSTAZIONI IN UN PRODOTTO FINALE!

6.2 Popolazione della base di dati

6.3 Connessione alla base di dati

6.4 Preparazione iniziale

6.5 Cliente

...

6.6 Disconnessione dalla base dei dati

