

# Progetto DADAU

## Disinfettazione Agile Di Aree Urbane

---

Gestione di campagne di disinfettazione agile di aree urbane (strade, mezzi, ecc.) in situazioni di emergenza socio-sanitaria

Realizzazione a cura del gruppo n.21:

Riccio Vincenzo - 0612703920

Sorrentino Giancarlo - 0612704169

Trapanese Lia - 0612704183

Triuzzi Emanuele - 0612704000

## **Introduzione**

La realtà che siamo costretti ad affrontare attualmente è purtroppo caratterizzata da situazioni contingenti, nazionali ed internazionali, certamente non semplici. Alla luce della persistente e diffusa emergenza socio-sanitaria, oltretutto imprevedibile, risulta di vitale importanza attuare in tempi brevi e in maniera efficace ogni tipo di soluzione, nonché predisporre contromisure opportune, capaci di fronteggiare al meglio le molteplici e crescenti problematiche.

Tra le tante necessità da affrontare a livello nazionale, il progetto DADAU si colloca nei contesti in cui è richiesta la gestione e coordinazione agile di campagne di disinfezione urbana, che necessitano la mobilitazione di squadre di volontari e professionisti, al fine di eseguire interventi mirati nelle aree urbane su tutto il territorio.

Tali interventi sono quindi destinati a coprire specifiche zone circoscritte all'interno dell'area di competenza, individuate dall'ente comunale o dai coordinatori delle campagne in cui si collocano, e prevede l'utilizzo di mezzi dotati di strumenti adatti alla disinfezione.

L'organizzazione e la gestione delle operazioni menzionate richiedono un grado elevato di efficacia ed efficienza, adeguato alle situazioni di emergenza in cui esse si svolgono, requisito fondamentale per il sistema informativo in cui il progetto DADAU mira a collocarsi.

## **Contesto generale**

Gestire le emergenze è sempre complicato in quanto spesso ci si trova in situazioni impreviste e singolari. È perciò necessario essere pronti a reagire efficacemente affinché si contengano i danni e si governi il problema.

Di fondamentale importanza risulta la ricerca degli strumenti atti allo scopo e la mobilitazione di risorse materiali ed umane. Nello specifico della trattazione di disinfezione di aree urbane in condizioni di emergenza socio-sanitaria, risulta necessaria la disponibilità di specialisti, volontari e mezzi appropriati.

Tale procedura è volta alla distruzione di microrganismi agenti eziologici di malattie infettive per impedirne la persistenza e la diffusione nell'ambiente, tutelando gli abitanti.

## **Obiettivo generale**

Questo progetto ha come obiettivo la realizzazione di una base di dati di un sistema informativo più ampio, in grado di monitorare gli interventi per la procedura di sanificazione nell'area urbana considerata e gestire l'allocazione delle risorse necessarie, quali volontari, professionisti e mezzi specifici per la sanificazione.

Pertanto, la disinfezione avverrà per gradi, e cioè ogni giorno l'intervento dei team sarà scaglionato per zona specifica, sino all'esaurimento della procedura.

## **Contesto specifico**

Coordinatori e addetti devono essere in grado sia di monitorare e gestire l'assegnazione delle risorse di ciascuna delle campagne già pianificate, sia di predisporre di nuove.

La base di dati che si intende realizzare si propone quindi di rappresentare i dati di ogni campagna di disinfezione, intesa come azione di sanificazione dell'intera area di interesse al fine di arginare il contagio e debellare i microrganismi.

I partecipanti, identificati dal codice fiscale, sono caratterizzati da un nome e un cognome, una data di nascita e uno o più numeri di telefono. Si vuole poi specificare se ciascun partecipante sia un impiegato o un volontario.

I mezzi coinvolti sono identificati da una targa e si distinguono in base alle loro caratteristiche, ovvero in base alle azioni di sanificazione che possono svolgere, e alla loro capacità, ovvero al massimo numero di passeggeri che possono trasportare.

Le zone da coprire sono identificate da un codice e ognuna di esse è costituita da più strade nelle quali verrà svolto l'intervento di disinfezione. Queste zone saranno quelle di riferimento per tutte le campagne organizzate mediante questo sistema informativo.

Gli interventi hanno una data di inizio, una data di fine, una zona da coprire, i mezzi utilizzati e un gruppo di partecipanti coinvolti.

Le campagne, identificate da un codice, hanno una data di inizio, una data di fine e comprendono uno o più interventi. Ogni campagna ha un proprio coordinatore, che deve necessariamente essere un impiegato.

## **Obiettivo realizzativo**

Il principale obiettivo realizzativo del progetto è quindi quello di fornire supporto a un'applicazione digitale, finalizzata al coordinamento di campagne di disinfezione agile, mediante l'utilizzo di un'architettura di tipo client-server e una base di dati, progettata ad hoc per tale scopo e realizzata tramite Postgres. Ciò al fine di fornire una tecnologia sicura e solida a supporto dello stato di emergenza in cui tali campagne si svolgono, grazie ai benefici che l'uso di un DBMS comporta: primi tra tutti la garanzia di un accesso rapido e sicuro ai dati e la presenza di vincoli di integrità sugli stessi, che vanno resi conformi alla realtà di interesse per garantire maggior controllo delle operazioni di sanificazione da effettuare e prevenire inconsistenze, le quali risulterebbero catastrofiche in situazioni critiche come quella presa in esame.

## Descrizione della realtà

La realtà d'interesse in cui è contestualizzato il progetto DADAU è un'emergenza socio-sanitaria che comporta la necessità di organizzare ed effettuare in tempi brevi campagne di disinfezione di aree urbane.

Il progetto può essere applicato in un qualunque ente locale territoriale autonomo ed è preferibile che questo abbia in forma accessibile uno stradario utile alla gestione e definizione delle zone su cui effettuare gli interventi che costituiscono una campagna.

Il territorio da disinfettare è quindi suddiviso in zone, ciascuna costituita da un gruppo di strade.

Le zone possono essere riutilizzabili tra differenti campagne riducendo così i tempi necessari ad avviarne una nuova. Per garantire flessibilità è possibile definire nuove zone in base alle esigenze.

Ciascuna campagna è caratterizzata da un codice identificativo, una descrizione, un responsabile, una data d'inizio e, se conclusa, una data di fine.

Ogni intervento di una campagna è svolto da un gruppo di partecipanti, che possono essere impiegati o volontari, che opera tramite mezzi in una determinata zona, con una data di inizio e di fine prestabilita.

Dei mezzi disponibili si vogliono conservare informazioni come targa, numero massimo di passeggeri e le loro caratteristiche, al fine di poter individuare i mezzi più adatti a ciascun intervento.

Di ogni partecipante è importante conoscere i principali dati anagrafici come codice fiscale, nome, cognome e data di nascita, uno o più numeri di telefono e la posizione che occupa (impiegato o volontario). Nel caso del volontario deve essere specificata anche la sua professione. Nel caso di impiegato, la sua data di assunzione e la relativa matricola.

## Tabella volumi

In riferimento a un comune di 150.000 abitanti

Campagne attive	3
Interventi	20 per campagna attiva
Mezzi	20
Zone	40
Strade	2500
Partecipanti	500

## **Principali operazioni**

- Aggiunta e modifica di una zona
- Definizione di una campagna
- Aggiunta di un intervento a una campagna esistente
- Aggiunta e modifica di una caratteristica di un mezzo
- Lettura delle informazioni relative alle campagne in corso (interventi programmati, zone coperte, personale e mezzi coinvolti)
- Ricerca degli interventi in corso
- Ricerca degli interventi che hanno coperto una zona
- Ricerca degli interventi che hanno coperto una zona nell'ambito di una specifica campagna
- Ricerca dei mezzi disponibili in una data
- Lettura delle informazioni anagrafiche relative al personale
- Ricerca degli interventi che coinvolgono un certo partecipante

## **Vincoli aziendali**

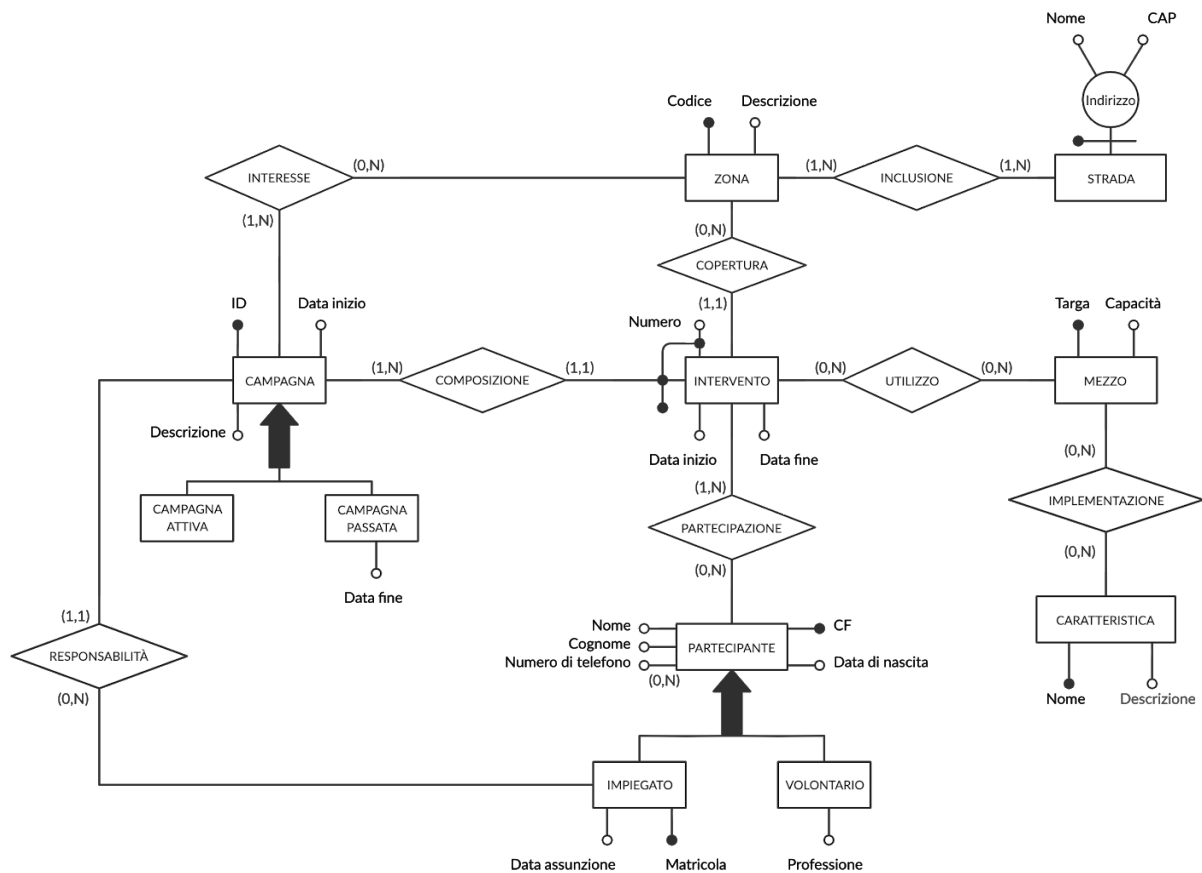
- Un mezzo deve essere coinvolto in un solo intervento alla volta
- Una persona deve essere coinvolta in un solo intervento alla volta
- Un intervento può coprire esclusivamente una delle zone associate alla campagna di cui fa parte
- Ogni data di fine non può essere antecedente alla data di inizio
- Il responsabile di una campagna non deve essere un volontario

## Bibliografia di riferimento

Basi di dati - P. Atzeni, R. Torlone, S. Paraboschi, S. Ceri, P. Fraternali

Materiale didattico fornito dai docenti M. Gaeta e A. D'Acerno

## Schema ER



## Strategia di Progetto

In seguito ad un approfondito studio della realtà di interesse è seguito un approccio di tipo inside-out, partendo dai concetti più importanti quali intervento e campagna per poi raggiungere i più periferici, come le caratteristiche di un mezzo.

## Glossario dei termini

Termine	Descrizione	Sinonimi	Collegamenti
Campagna	Insieme di uno o più interventi che hanno un obiettivo comune		Responsabile Intervento Zona
Intervento	Messa in atto in una determinata zona, con determinati mezzi e persone, dell'azione necessaria al raggiungimento dell'obiettivo previsto dalla campagna	Azione, Operazione	Campagna Persona Zona Mezzo
Zona	Raggruppamento di strade effettuato secondo criteri ben precisi e che è oggetto di un intervento	Area di interesse	Strada Campagna Intervento
Partecipante	Persona che partecipa (o intende partecipare) agli interventi	Persona	Intervento
Impiegato	Partecipante che lavora presso l'ente che gestisce le campagne		Campagna
Volontario	Partecipante che non lavora presso l'ente ma si offre disponibile per partecipare a uno o più interventi		
Mezzo	Veicolo dotato di specifiche attrezzature necessarie alla messa in atto dell'intervento		Intervento
Caratteristica	Peculiarità di un mezzo, utile per la scelta di un mezzo appropriato per l'intervento	Peculiarità	

## Operazioni

- **Op1:** Aggiunta di un intervento a una campagna attiva
- **Op2:** Lettura di tutti gli interventi di una campagna, con la relativa zona coperta
- **Op3:** Lettura del numero di interventi che hanno coperto le zone di interesse in una campagna
- **Op4:** Ricerca dei partecipanti coinvolti in un intervento
- **Op5:** Ricerca dei mezzi utilizzati in un intervento
- **Op6:** Ricerca della zona coperta da un intervento e delle relative strade
- **Op7:** Ricerca degli interventi in corso
- **Op8:** Ricerca dei mezzi disponibili in un dato intervallo temporale
- **Op9:** Ricerca di partecipanti disponibili in un dato intervallo temporale
- **Op10:** Lettura delle informazioni di un partecipante



## Tavola delle operazioni

L'analisi delle seguenti operazioni fa riferimento al periodo di attività delle campagne.

Operazione	Tipo	Frequenza
Aggiunta di un intervento alle campagne attive	I	30 a settimana
Lettura di tutti gli interventi di una campagna, con relativa zona coperta	I	10 a settimana
Lettura del numero di interventi che hanno coperto le zone di interesse in una campagna	I	20 a settimana
Ricerca dei partecipanti coinvolti in un intervento	I	30 a settimana
Ricerca dei mezzi utilizzati in un intervento	I	25 a settimana
Ricerca della zona coperta da un intervento e delle relative strade	I	15 a settimana
Ricerca degli interventi in corso	B	20 a settimana
Ricerca dei mezzi disponibili in una dato intervallo temporale	I	40 a settimana
Ricerca di partecipanti disponibili in un dato intervallo temporale	I	30 a settimana
Lettura delle informazioni di un partecipante	I	35 a settimana

## Tavola dei volumi

Concetto	Tipo	Volume
Campagna	E	15
Campagna attiva	E	3
Campagna passata	E	12
Zone	E	40
Strade	E	2500
Interventi	E	$20 * 15 = 300$
Partecipante	E	500
Impiegato	E	400
Volontario	E	100
Mezzo	E	20
Responsabilità	R	15
Interesse	R	$15 * 15 = 225$
Inclusione	R	$2500 * 3 = 7500$
Copertura	R	300
Composizione	R	300
Partecipazione	R	$300 * 4 = 1200$
Utilizzo	R	$300 * 1.1 = 330$

## Tavole degli accessi

Analisi degli accessi per ogni operazione

### Op1: Aggiunta di un intervento a una campagna attiva

Descrizione: si ipotizza l'aggiunta di un intervento a una campagna attiva esistente.

Concetto	Costrutto	Accessi	Tipo
Intervento	E	1	S
Partecipante	E	4	L
Partecipazione	R	4	S
Mezzo	E	1	L
Utilizzo	R	1	S
Zona	E	1	L
Copertura	R	1	S
Campagna attiva	E	1	L
Composizione	R	1	S
Costo totale:			27

### Op2: Lettura di tutti gli interventi di una campagna, con la relativa zona coperta

Descrizione: Si ipotizza di voler ottenere una lista di tutti gli interventi di una campagna, comprensivi della relativa zona coperta.

Concetto	Costrutto	Accessi	Tipo
Campagna	E	1	L
Composizione	R	20	L
Intervento	E	20	L
Copertura	R	20	L
Zona	E	20	L
Costo totale:			81

**Op3: Lettura del numero di interventi che hanno coperto le zone di interesse in una campagna**

Descrizione: Si ipotizza di voler sapere quante volte le zone di interesse nell'ambito di una campagna siano state coperte dai relativi interventi. Utile per conoscere le zone che devono essere coperte in una campagna ma ancora non lo sono state.

Concetto	Costrutto	Accessi	Tipo
Campagna	E	1	L
Composizione	R	20	L
Intervento	E	20	L
Copertura	R	20	L
Zona	E	20	L
Costo totale:			81

**Op4: Ricerca dei partecipanti coinvolti in un intervento**

Descrizione: si ipotizza la ricerca dei partecipanti a un intervento.

Concetto	Costrutto	Accessi	Tipo
Intervento	E	1	L
Partecipazione	R	4	L
Partecipante	E	4	L
Costo totale:			9

**Op5: Ricerca dei mezzi utilizzati in un intervento**

Descrizione: si ipotizza la ricerca dei mezzi utilizzati in un intervento.

Concetto	Costrutto	Accessi	Tipo
Intervento	E	1	L
Utilizzo	R	1	L
Mezzo	E	1	L
Costo totale:			3

**Op6: Ricerca della zona coperta da un intervento con le relative strade**

Descrizione: si ipotizza la ricerca della zona coperta da un intervento e la lista delle strade incluse in quella zona.

Concetto	Costrutto	Accessi	Tipo
Intervento	E	1	L
Copertura	R	1	L
Zona	E	1	L
Inclusione	R	62	L
Strada	E	62	L
Costo totale:			127

**Op7: Ricerca degli interventi in corso**

Descrizione: si ipotizza di voler avere un quadro generale degli interventi in corso. Si suppone che in media non vi siano più di due interventi in corso contemporaneamente in una stessa campagna attiva.

Concetto	Costrutto	Accessi	Tipo
Campagna attiva	E	3	L
Composizione	R	5	L
Intervento	E	5	L
Costo totale:			13

**Op8: Ricerca dei mezzi disponibili in un dato intervallo temporale**

Descrizione: si ipotizza di voler conoscere i mezzi disponibili in un dato intervallo temporale.

Si suppone un intervallo temporale pari alla durata media di un intervento, dato che questa operazione ha lo scopo di fornire informazioni sui mezzi utilizzabili nello stesso, ai fini dell'allocazione di un nuovo intervento.

Poiché si assume sia raro che ci possano essere più di due interventi in corso contemporaneamente per ogni campagna attiva, e che in media ogni intervento fa utilizzo di un solo mezzo, la relazione utilizzo verrà letta in media  $3 \cdot 2 \cdot 1.1 = 7$  volte.

Concetto	Costrutto	Accessi	Tipo
Campagna attiva	E	3	L
Composizione	R	60	L
Intervento	E	60	L
Utilizzo	R	7	L
Mezzo	E	20	L
Costo totale:			160

#### Op9: Ricerca di partecipanti disponibili in un dato intervallo temporale

Descrizione: si ipotizza di voler conoscere i partecipanti disponibili in un dato intervallo temporale.

Si suppone un intervallo temporale pari alla durata media di un intervento. Poiché si assume rara la possibilità di avere più di due interventi in corso contemporaneamente per ogni campagna attiva, e che in media ad ogni intervento partecipino 4 persone, la relazione partecipazione verrà letta in media solo  $3 \cdot 2 \cdot 4 = 24$  volte.

Concetto	Costrutto	Accessi	Tipo
Campagna attiva	E	3	L
Composizione	R	60	L
Intervento	E	60	L
Partecipazione	R	24	L
Partecipante	E	500	L
Costo totale:			648

#### Op10: Lettura delle informazioni di un partecipante

Descrizione: si ipotizza di voler conoscere i dettagli su un dato partecipante.

Concetto	Costrutto	Accessi	Tipo
Partecipante	E	1	L
Costo totale:			1

## Dizionario dei dati dello schema

Entità	Descrizione	Attributi	Identificatore
Campagna	Insieme di operazioni con un determinato fine	ID, descrizione, data inizio	ID
Campagna attiva	Specializzazione di Campagna. Sono presenti interventi ancora da svolgere o programmare	Ereditati da campagna: ID, descrizione, data inizio	ID
Campagna passata	Specializzazione di Campagna. Tutti gli interventi sono terminati e non sono previsti ulteriori interventi	Ereditati da Campagna: ID, descrizione, data inizio Data fine	ID
Intervento	Operazione di disinfezione localizzata in una zona	Campagna, numero, data inizio, data fine	Campagna, numero
Partecipante	Persona coinvolta o disponibile in una o più campagne	CF, nome, cognome, data di nascita, disponibile, numero di telefono	CF
Impiegato	Persona che presta attività professionale	Ereditati da Partecipante: CF, nome, cognome, data di nascita, disponibile, numero di telefono (attributo multivalore) Matricola	Matricola
Volontario	Dipendente dalla volontà individuale del singolo	Ereditati da Partecipante: CF, nome, cognome, data di nascita, numero di telefono Professione volontario	CF

Strada	Strada di un comune	Indirizzo (attributo composto): Nome, CAP	Indirizzo (attributo composto): Nome, CAP
Zona	Insieme di strade	Codice, descrizione	Codice
Mezzo	Mezzo di trasporto adibito a svolgere un intervento	Targa, capacità, disponibile	Targa
Caratteristica	Indica le funzionalità che possono essere implementate da un mezzo	Nome, descrizione	

Relazione	Descrizione	Entità coinvolte	Attributi
Responsabilità	Associa una campagna al suo responsabile	Campagna (1,1) Impiegato (0,N)	
Interesse	Definisce le zone d'interesse di una campagna	Campagna (1,N) Zona (0,N)	
Composizione	Indica gli interventi da cui è composta una campagna.	Campagna (1,N) Intervento (1,1)	
Copertura	Specifica la zona in cui è svolto l'intervento	Intervento (1,1) Zona (0,N)	
Inclusione	Indica le strade contenute in una zona	Zona (1,N) Strada (1,N)	
Utilizzo	Indica i mezzi coinvolti in un intervento	Intervento (0,N) Mezzo (0,N)	
Implementazione	Specifica le caratteristiche implementate dai mezzi	Mezzo (0,N) Caratteristica (0,N)	
Partecipazione	Indica le persone coinvolte in un intervento	Intervento (1,N) Partecipante (0,N)	



## Regole aziendali per lo schema

Regole di vincolo	
<b>RV1</b>	La data di fine di una campagna deve essere successiva alla data d'inizio della campagna stessa e a quella conclusiva di ogni intervento che la compone.
<b>RV2</b>	Non è possibile rimuovere gli interventi delle campagne già concluse. Allo stesso modo non è possibile modificare le relative zone di interesse. Non è inoltre possibile rimuovere le campagne in cui è stato effettuato almeno un intervento.
<b>RV3</b>	Un intervento non può iniziare prima della campagna attiva a cui fa riferimento. Al contempo, esso non può terminare prima della sua data d'inizio.
<b>RV4</b>	L'aggiunta di un intervento ad una campagna è permesso se la zona che copre è presente tra le zone d'interesse della campagna.
<b>RV6</b>	Una campagna si può concludere solo se non presenta interventi non conclusi e se è stato effettuato almeno un intervento per ogni zona di interesse della campagna stessa.
<b>RV7</b>	Non è possibile rimuovere una zona d'interesse di una campagna se vi è già stato effettuato almeno un intervento.
<b>RV8</b>	Non è possibile impegnare partecipanti e/o mezzi in un intervento se essi non sono disponibili o già impegnati nell'intervallo temporale in cui esso si svolge.

## Qualità dello schema

Dall'analisi dello schema frutto della progettazione concettuale è emerso che esso rispetta i requisiti di correttezza, in quanto fa un appropriato utilizzo dei costrutti del modello E-R, e di completezza, in quanto tutte le specifiche sono state rappresentate mediante lo schema e tutte le operazioni previste sono realizzabili navigando attraverso lo schema. Inoltre, rientra nei canoni inerenti la leggibilità, in quanto la posizione delle entità nello schema è condizionata dal numero di associazioni di cui le stesse fanno parte.

Infine, il requisito di minimalità è rispettato, in quanto non sono presenti ridondanze, mentre per quanto concerne la presenza dei cicli, essi sono necessari come da specifiche e non rappresentano alcun tipo di ridondanza.

## Progettazione Logica e ristrutturazione dello schema Entità-Relazione

Terminata la fase concettuale di progettazione, si passa alla progettazione logica: viene ristrutturato lo schema E-R e tradotto nel modello logico scelto, nel nostro caso il relazionale.

Ne consegue quindi l'operazione di normalizzazione e l'implementazione in linguaggio SQL, la quale si occuperà di garantire consistenza con lo schema indicato, le specifiche di progetto e le regole aziendali espresse.

All'inizio dell'operazione di ristrutturazione è stata effettuata l'analisi delle ridondanze. Durante la stessa è risultato che nelle operazioni prese in considerazione durante la fase concettuale non sono presenti ridondanze da analizzare. Tuttavia, è risultato opportuno introdurre una ridondanza, in particolare l'attributo "Numero Coperture" sulla relazione "Interesse", che indica il numero di interventi effettuati per ogni zona di interesse di ogni campagna. Ciò al fine di diminuire notevolmente il numero di accessi effettuati durante l'operazione 3, la cui tavola delle operazioni con l'aggiunta della ridondanza è riportata di seguito:

Concetto	Costrutto	Accessi	Tipo
Campagna	E	1	L
Interesse	R	15	L
Costo totale:			16

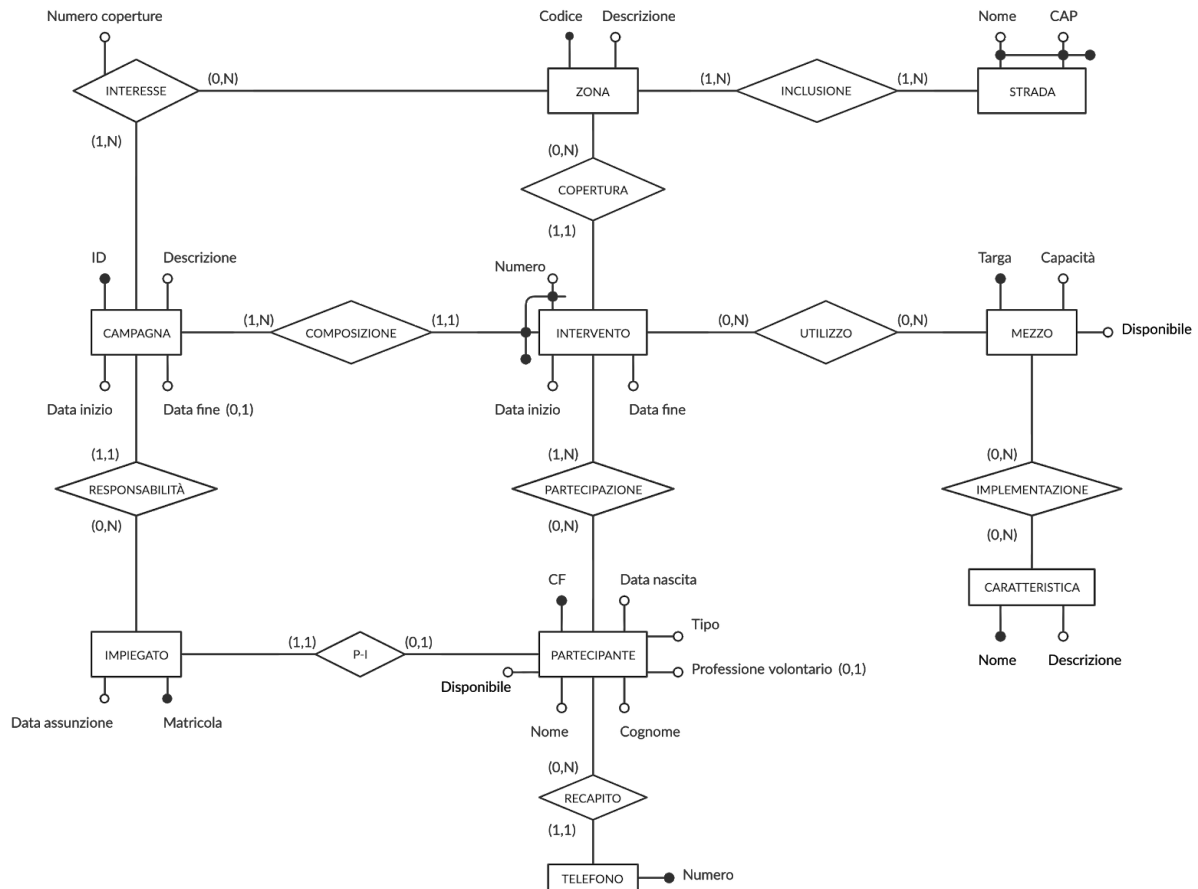
Il numero di accessi è quindi diminuito da 81 a 16.

Per quanto concerne la rimozione delle generalizzazioni, sono stati utilizzati due approcci diversi per quelle presenti all'interno dello schema concettuale iniziale.

In particolare, la generalizzazione riguardante l'entità Campagna è stata eliminata sfruttando l'approccio che prevede l'accorpamento delle entità figlie in quella genitore. È stato quindi aggiunto l'attributo "Data fine" alla suddetta entità che, se NULL, indica che la campagna è attiva; viceversa, che è terminata.

Per la generalizzazione riguardante l'entità partecipante è stato invece scelto un approccio di tipo ibrido: l'entità Volontario è stata accorpata in Partecipante, mentre l'entità Impiegato è stata reificata ed associata all'entità Partecipante. A quest'ultima sono stati quindi aggiunti gli attributi "Tipo", che indica se il partecipante è un impiegato o un volontario, e "Professione", rinominato in "Professione volontario" per questioni di leggibilità dello schema, ereditato da Volontario, che assume valore NULL se il partecipante è un impiegato.

Infine, come si evince dallo schema ER ristrutturato, per la scelta degli identificatori non è stato necessario alcun intervento, in quanto quelli indicati nel diagramma si sono rivelati adeguati allo scopo.



## Traduzione Logica

Per ogni relazione indicata di seguito, le quali sono il risultato dell'applicazione dei pattern della traduzione verso il modello relazionale dello schema precedentemente ristrutturato, la chiave è sottolineata e sono identificati con un asterisco gli attributi che possono accettare un valore nullo.

**Strada** (CAP, Nome)

**Inclusione** (CAP, Nome, Zona)

**Zona** (Codice, Descrizione)

**Interesse** (Campagna, Zona, Numero\_Coperture)

**Campagna** (ID, Descrizione, Data\_Inizio, Data\_Fine\*, Responsabile)

**Intervento** (Campagna, Numero, Zona, Data\_Inizio, Data\_Fine)

**Partecipazione** (Campagna, Numero, Partecipante)

**Partecipante** (CF, Disponibile, Nome, Cognome, Data\_Nascita, Tipo, Professione\_Volontario)

**Impiegato** (Matricola, CF, Data\_Assunzione)

**Telefono** (Numero, Proprietario)

**Utilizzo** (Campagna, Numero, Targa)

**Mezzo** (Targa, Disponibile, Capacità)

**Implementazione** (Targa, Caratteristica)

**Caratteristica** (Nome, Descrizione)

Di seguito sono quindi riportati i vincoli di foreign key delle suddette relazioni.

### Vincoli di foreign key

Inclusione(Zona) REFERENCES Zona(Codice);

Inclusione(CAP, nome) REFERENCES Strada(CAP, nome);

Interesse(Campagna) REFERENCES Campagna(ID);

Interesse(Zona) REFERENCES Zona(Codice);

Campagna(Responsabile) REFERENCES Impiegato(Matricola);

Intervento(Campagna) REFERENCES Campagna(ID);

Intervento(Zona) REFERENCES Zona(Codice);

Partecipazione(Partecipante) REFERENCES Partecipante(CF);

Partecipazione(Campagna, Numero) REFERENCES Intervento(Campagna, Numero);

Impiegato(CF) REFERENCES Partecipante(CF);

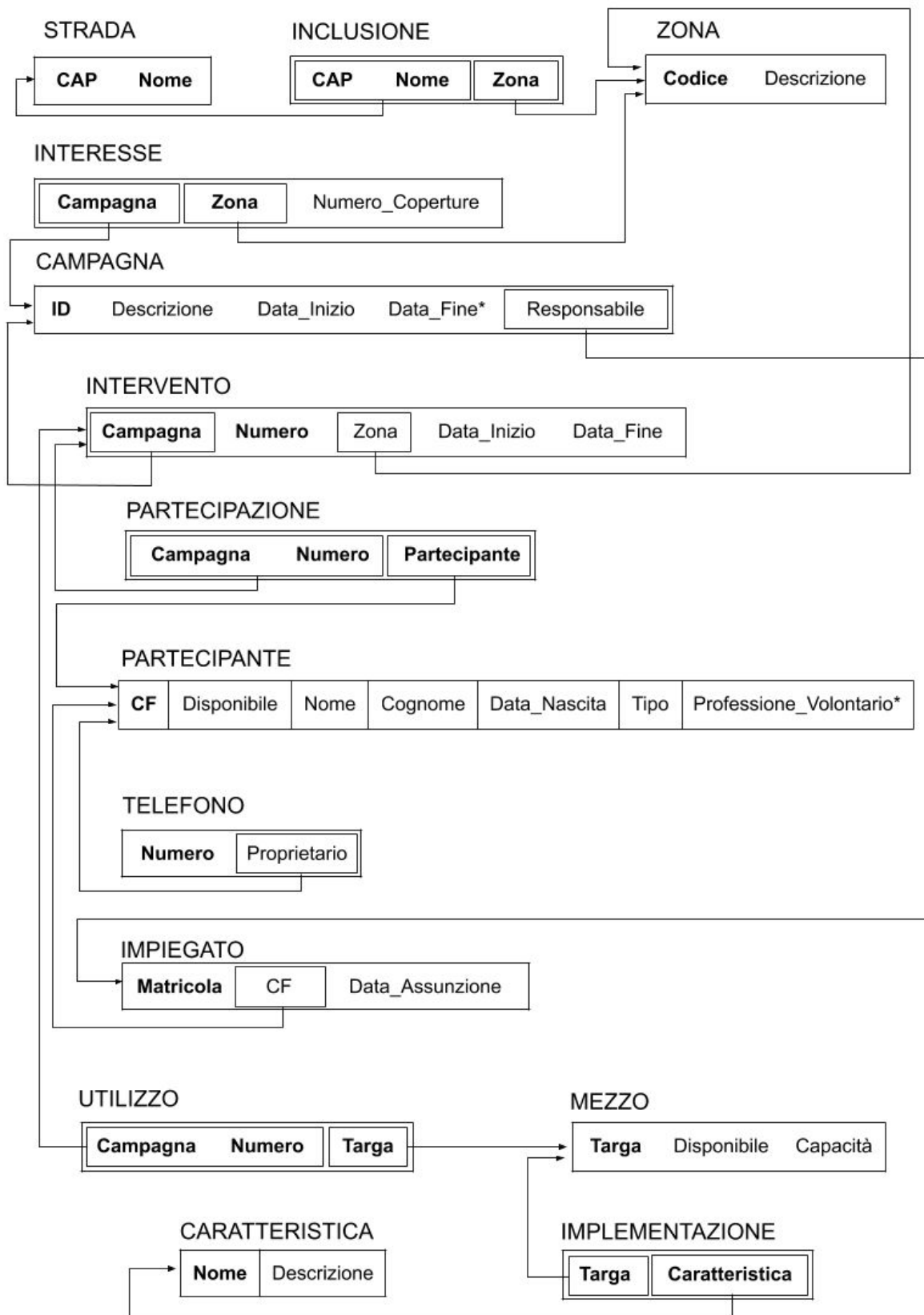
Telefono(Proprietario) REFERENCES Partecipante(CF);

Utilizzo(Targa) REFERENCES Mezzo(Targa);

Utilizzo(Campagna, Numero) REFERENCES Intervento(Campagna, Numero);

Implementazione(Targa) REFERENCES Mezzo(Targa);

Implementazione(Caratteristica) REFERENCES Caratteristica(Nome);



## Normalizzazione

Considerando la definizione di forma normale di Boyce-Codd, è facilmente verificabile, per ogni relazione, che tutte le dipendenze funzionali non banali hanno come determinante una superchiave della relazione. Ne consegue quindi che le relazioni sono anche in terza forma normale.

### Dipendenze funzionali non banali:

#### **Impiegato(Matricola, CF, Data\_Assunzione):**

Matricola  $\rightarrow$  CF, Data\_Assunzione

CF  $\rightarrow$  Matricola, Data\_Assunzione

Sia Matricola che CF sono una chiave per la relazione, per cui è in BCNF.

Tutte altre dipendenze funzionali non sono che vincoli di chiave, per cui verificano automaticamente la BCNF.

## Dataset

Le seguenti sono alcune delle query per il popolamento del DB creato. Lo script completo è allegato al documento.

```
--Volontari
CALL insert_volontario('TRZMNL97P14H703P', 'Lorenzo', 'Triuzzi', '1997-07-06',
'ingegnere');
--Impiegati
CALL insert_impiegato('RCCVCN98S16H703M', '0612703920', 'Vincenzo', 'Riccio',
'1998-11-16', '2017-10-01');
CALL insert_impiegato('TRPLIA98M38H703N', '0612704183', 'Lia', 'Trapanese',
'1998-10-06', '2017-10-01');
CALL insert_impiegato('TRZMNL99P14H703P', '0612704000', 'Emanuele', 'Triuzzi',
'1999-09-14', '2017-10-01');
CALL insert_impiegato('SRRGCR99B11C361H', '0612704179', 'Giancarlo', 'Sorrentino',
'1999-02-11', '2017-10-01');
--Numeri di telefono
INSERT INTO telefono (numero, proprietario) VALUES ('3484315206',
'SRRGCR99B11C361H');
--Strada
INSERT INTO strada (nome,cap) VALUES ('Viale Alfonso Tesauro', 84127);
INSERT INTO strada (nome,cap) VALUES ('Via Sabato Robertelli', 84127);
INSERT INTO strada (nome,cap) VALUES ('Viale Richard Wagner', 84131);
--Zona
INSERT INTO zona(codice,descrizione) VALUES ('TOR-01', 'zona Torrione alto');
INSERT INTO zona(codice,descrizione) VALUES ('TOR-02', 'zona Torrione a livello del
mare');
INSERT INTO zona(codice,descrizione) VALUES ('ARB-00', 'zona di Arbostella');
--Inclusione
INSERT INTO inclusione(nome,cap,zona) VALUES ('Viale Richard Wagner', 84131,
'ARB-00');
INSERT INTO inclusione(nome,cap,zona) VALUES ('Via Sabato Robertelli', 84127,
'TOR-02');
INSERT INTO inclusione(nome,cap,zona) VALUES ('Viale Alfonso Tesauro', 84127,
'TOR-01');
--Mezzo
INSERT INTO mezzo(targa, capacità, disponibile) VALUES ('AA000AA',1,TRUE);
--Caratteristica
INSERT INTO caratteristica(nome) VALUES ('elevazione');
--Equipaggiamento
INSERT INTO implementazione(targa, caratteristica) VALUES ('AA000AA','elevazione');
INSERT INTO implementazione(targa, caratteristica) VALUES ('BB000BB','elevazione');
```

```

--Campagna
INSERT INTO campagna(id, descrizione, data_inizio, responsabile)
VALUES ('Covid19-1','Descrizione covid19-1',LOCALTIMESTAMP,'0612704000');
INSERT INTO campagna(id, descrizione, data_inizio, responsabile)
VALUES ('Covid19-2','Legambiente Covid',LOCALTIMESTAMP,'0612704179');

--Interesse
INSERT INTO interesse(campagna,zona) VALUES ('Covid19-1','TOR-01');
INSERT INTO interesse(campagna,zona) VALUES ('Covid19-1','TOR-02');
INSERT INTO interesse(campagna,zona) VALUES ('Covid19-2','ARB-00');

--Interventi
INSERT INTO intervento(campagna,zona,data_inizio,data_fine)
VALUES ('Covid19-1', 'TOR-01', LOCALTIMESTAMP, LOCALTIMESTAMP + INTERVAL '10
hour');
INSERT INTO intervento(campagna,zona,data_inizio,data_fine)
VALUES ('Covid19-1', 'TOR-02', LOCALTIMESTAMP + INTERVAL '1 hour', LOCALTIMESTAMP +
INTERVAL '3 hour');
INSERT INTO intervento(campagna,zona,data_inizio,data_fine)
VALUES ('Covid19-2', 'ARB-00', LOCALTIMESTAMP, LOCALTIMESTAMP + INTERVAL '10
hour');

--Partecipazione
INSERT INTO partecipazione (campagna, numero, partecipante) values ('Covid19-1', 1,
'TRZMNL99P14H703P');
INSERT INTO partecipazione (campagna, numero, partecipante) values ('Covid19-1', 2,
'SRRGCR99B11C361H');
INSERT INTO partecipazione (campagna, numero, partecipante) values ('Covid19-2', 1,
'RCCVCN98S16H703M');

--Utilizzo
INSERT INTO utilizzo(campagna, numero, targa) VALUES ('Covid19-1', 1, 'AA000AA');
INSERT INTO utilizzo(campagna, numero, targa) VALUES ('Covid19-1', 1, 'BB000BB');
INSERT INTO utilizzo(campagna, numero, targa) VALUES ('Covid19-1', 2, 'CC000CC');

```



## Query sviluppate

### Procedura per inserire volontari e impiegati

Per inserire un impiegato nella tabella è necessario che questo, per vincolo, sia presente anche nella tabella partecipante. È necessaria una procedura poiché un partecipante di tipo impiegato, per essere inserito, deve essere presente in impiegato. Il vincolo di foreign key viene verificato solo al termine della procedura.

```
CREATE OR REPLACE PROCEDURE insert_impiegato(cf T_CF, matricola T_MATRICOLA, nome
VARCHAR(20), cognome VARCHAR(20), data_nascita DATE, data_assunzione DATE)
AS $$
    SET CONSTRAINTS FK_cf DEFERRED;
    INSERT INTO impiegato(matricola, cf, data_assunzione)
    VALUES (matricola, cf, data_assunzione);
    INSERT INTO partecipante(cf, nome, cognome, data_nascita, tipo, disponibile)
    VALUES (cf, nome, cognome, data_nascita, 'impiegato', true);
$$
LANGUAGE SQL;
```

La seguente, non è strettamente necessaria, è stata inserita per comodità e simmetria vista la necessità della precedente per l'inserimento di un impiegato.

```
CREATE OR REPLACE PROCEDURE insert_volontario(cf T_CF, nome VARCHAR(20), cognome
VARCHAR(20), data_nascita DATE, professione VARCHAR(15))
AS $$
    INSERT INTO partecipante(cf, nome, cognome, data_nascita, tipo,
professione_volontario)
    VALUES (cf, nome, cognome, data_nascita, 'volontario', professione);
$$
LANGUAGE SQL;
```

Di seguito sono riportate le operazioni principali indicate precedentemente.

### Operazione 1: Aggiunta di un intervento ad una campagna

```
INSERT INTO intervento(campagna, zona, data_inizio, data_fine)
VALUES ('NOME CAMP', 'ZON-NN', LOCALTIMESTAMP, LOCALTIMESTAMP + INTERVAL '10 hour');
```

## Operazione 2: Lettura di tutti gli interventi di una campagna, con la relativa zona coperta

```
CREATE OR REPLACE FUNCTION leggi_interventi (id_camp T_ID_CAMP)
RETURNS TABLE (numero_intervento UNSIGNED_INT, zona T_CODICE_ZONA, data_inizio
TIMESTAMP, data_fine TIMESTAMP) AS $$
BEGIN
    RETURN QUERY
        SELECT I.numero, CAST (I.zona as T_CODICE_ZONA), I.data_inizio,
I.data_fine
        FROM intervento as I
        WHERE I.campagna = id_camp;
END;
$$
LANGUAGE 'plpgsql';
```

## Operazione 3: Lettura del numero di interventi che hanno coperto le zone di interesse di una specifica campagna

```
CREATE OR REPLACE FUNCTION num_cop_zone_interesse (id_camp T_ID_CAMP)
RETURNS TABLE (zona T_CODICE_ZONA, numero_di_interventi UNSIGNED_INT) AS $$
BEGIN
    RETURN QUERY
        SELECT CAST (I.zona as T_CODICE_ZONA), I.numero_coperture
        FROM interesse as I
        WHERE I.campagna = id_camp;
END;
$$
LANGUAGE 'plpgsql';
```

## Operazione 4: Ricerca dei partecipanti coinvolti in uno specifico intervento

```
CREATE OR REPLACE FUNCTION partecipanti_intervento (id_camp T_ID_CAMP,
num_intervento UNSIGNED_INT)
RETURNS TABLE (tipo VARCHAR(10), CF T_CF, nome VARCHAR(20), cognome VARCHAR(20)) AS
$$
BEGIN
    RETURN QUERY
        SELECT partecipante.tipo, CAST (partecipante.CF as T_CF),
partecipante.nome, partecipante.cognome
        FROM partecipazione
        JOIN partecipante ON partecipazione.partecipante = partecipante.CF
```

```

        WHERE partecipazione.campagna = id_camp AND partecipazione.numero =
num_intervento;
    END;
$$
LANGUAGE 'plpgsql';

```

### Operazione 5: Ricerca dei mezzi utilizzati in uno specifico intervento

```

CREATE OR REPLACE FUNCTION mezzi_intervento (id_camp T_ID_CAMP, num_intervento
UNSIGNED_INT)
RETURNS TABLE (targa T_TARGA) AS $$
BEGIN
    RETURN QUERY
        SELECT CAST(utilizzo.targa AS T_TARGA)
        FROM utilizzo
        WHERE utilizzo.campagna = id_camp AND utilizzo.numero = num_intervento;
    END;
$$
LANGUAGE 'plpgsql';

```

### Operazione 6: Ricerca della zona e delle relative strade coperte da uno specifico intervento

```

CREATE OR REPLACE FUNCTION zona_strade_coperte (id_camp T_ID_CAMP, num_intervento
UNSIGNED_INT)
RETURNS TABLE (zona T_CODICE_ZONA, nome_strada VARCHAR(30), CAP T_CAP) AS $$
BEGIN
    RETURN QUERY
        SELECT CAST(inclusione.zona AS T_CODICE_ZONA), inclusione.nome,
CAST(inclusione.CAP AS T_CAP)
        FROM inclusione
        WHERE inclusione.zona = (SELECT intervento.zona FROM intervento WHERE
campagna = id_camp AND numero = num_intervento);
    END;
$$
LANGUAGE 'plpgsql';

```

### Operazione 7: Ricerca degli interventi in corso

```

CREATE OR REPLACE FUNCTION interventi_in_corso ()
RETURNS TABLE (id_camp T_ID_CAMP, num_intervento UNSIGNED_INT, data_fine TIMESTAMP)
AS $$
    DECLARE now TIMESTAMP := LOCALTIMESTAMP;
BEGIN
    RETURN QUERY

```

```

        SELECT intervento.campagna, intervento.numero, intervento.data_fine
        FROM intervento
        WHERE intervento.data_fine > now AND intervento.data_inizio < now;

    END;

$$

LANGUAGE 'plpgsql';

```

## Operazione 8: Ricerca dei mezzi disponibili in un dato intervallo temporale

```

CREATE OR REPLACE FUNCTION mezzi_disponibili (ora_inizio TIMESTAMP, ora_fine
TIMESTAMP)
RETURNS TABLE (targa T_TARGA, capacità UNSIGNED_INT, caratteristica VARCHAR(20)) AS
$$
BEGIN
    RETURN QUERY
        SELECT M.targa, M.capacità, M.caratteristica_implementata AS Caratteristica
        FROM mezzi_con_caratteristiche AS M
        WHERE M.disponibile AND M.targa NOT IN (
            --Ottengo tutti i mezzi utilizzati nell'intervallo di interesse
            SELECT U.Targa
            FROM intervento I
            JOIN utilizzo U
            ON (I.campagna = U.campagna AND I.numero = U.numero)
            WHERE (I.data_inizio BETWEEN ora_inizio AND ora_fine) OR (I.data_fine
BETWEEN ora_inizio AND ora_fine)
        )
        ORDER BY M.targa;

    END;

$$

LANGUAGE 'plpgsql';

```

## Operazione 9: Ricerca di partecipanti disponibili in un dato intervallo temporale

```

CREATE OR REPLACE FUNCTION partecipanti_disponibili (ora_inizio TIMESTAMP, ora_fine
TIMESTAMP)
RETURNS TABLE (Tipo VARCHAR(10), CF T_CF, Nome VARCHAR (20), Cognome VARCHAR(20),
Data_Di_Nascita DATE, Professione_Volontario VARCHAR(15)) AS $$
BEGIN
    RETURN QUERY
        SELECT P.tipo, P.CF, P.Nome, P.Cognome, P.Data_Nascita,
P.Professione_Volontario
        FROM partecipante P
        WHERE P.disponibile AND NOT EXISTS (

```

```

        SELECT PZ.partecipante
        FROM intervento I
        JOIN partecipazione PZ
        ON (I.campagna = PZ.campagna AND I.numero = PZ.numero)
        WHERE PZ.partecipante = P.CF AND I.data_inizio BETWEEN ora_inizio AND
ora_fine OR I.data_fine BETWEEN ora_inizio AND ora_fine
    )
    ORDER BY P.tipo, P.cognome;
END;
$$
LANGUAGE 'plpgsql';

```

## Operazione 10: Lettura delle informazioni relative di un partecipante

```

CREATE OR REPLACE FUNCTION lettura_partecipante (cf_par T_CF)
RETURNS TABLE (Tipo VARCHAR(10), CF T_CF, Disponibile BOOLEAN, Nome VARCHAR (20),
Cognome VARCHAR(20), Data_Di_Nascita DATE, Numero_Di_Telefono VARCHAR(15),
Matricola_Impiegato T_Matricola, Data_Assunzione_Impiegato DATE,
Professione_Volontario VARCHAR(15)) AS $$
BEGIN
RETURN QUERY
    SELECT *
    FROM dati_partecipanti P
    WHERE P.CF = cf_par;
END;
$$
LANGUAGE 'plpgsql';

```

## Query non banali

### Numero di interventi effettuati sulle zone coperte almeno due volte in ciascuna campagna

```

SELECT zona as Zona_Coperta, COUNT(zona) as interventi_effettuati, campagna
FROM intervento
GROUP BY campagna, zona
HAVING COUNT(zona)>=2
ORDER BY zona, campagna;

```

## Media del numero di partecipanti per ogni intervento che ha coinvolto almeno due mezzi di una campagna

```
SELECT R.campagna, CAST(AVG(R.numero_partecipanti) as numeric(10,2)) as
media_partecipanti
FROM (
    SELECT I.campagna AS Campagna, I.Numero AS Intervento, count(DISTINCT
P.partecipante) AS Numero_Partecipanti
    FROM intervento I, partecipazione P, utilizzo U
    WHERE I.campagna = P.campagna AND I.numero = P.Numero AND I.Campagna =
U.Campagna AND I.Numero = U.Numero
    GROUP BY I.campagna, I.numero
    HAVING count(DISTINCT U.targa)>=2
) AS R
GROUP BY R.campagna;
```

## Query insiemistica, restituisce tutte le zone che non sono ancora state coperte da un intervento di una specifica campagna

```
SELECT Z.codice, Z.descrizione
FROM interesse I
JOIN zona Z ON (I.zona = Z.codice)
WHERE I.campagna = 'DISINF-01'
EXCEPT
SELECT Z.codice, Z.descrizione
FROM intervento I
JOIN zona Z ON (I.zona = Z.codice)
WHERE I.Campagna = 'DISINF-01';
```

## Partecipante più attivo di una campagna

```
SELECT PC.id_campagna, P.cf, P.nome, P.cognome, PC.interventi_effettuati
FROM partecipazioni_campagna PC
JOIN partecipante P
ON PC.partecipante = P.cf
WHERE (id_campagna, interventi_effettuati) IN (
    -- numero più altro di interventi effettuati da un partecipante in ogni campagna
    SELECT id_campagna, MAX(interventi_effettuati)
    FROM partecipazioni_campagna
    GROUP BY id_campagna
);
```

## Viste prodotte

### Vista delle informazioni complete di ciascun impiegato

```
CREATE OR REPLACE VIEW dettagli_impiegati AS
    SELECT I.matricola, I.cf, P.nome, P.cognome, I.data_assunzione, P.data_nascita,
    P.disponibile
    FROM impiegato I
    JOIN partecipante P
    ON I.cf = P.cf;
```

### Vista del numero di interventi effettuati da ogni partecipante in ogni campagna

```
CREATE OR REPLACE VIEW partecipazioni_campagna (id_campagna, partecipante,
interventi_effettuati) AS
    SELECT campagna, partecipante, count(*) AS numero_partecipazioni
    FROM partecipazione
    GROUP BY campagna, partecipante
    ORDER BY numero_partecipazioni DESC;
```

### Vista dei mezzi e delle caratteristiche associate

```
CREATE OR REPLACE VIEW mezzi_con_caratteristiche(targa, disponibile, capacità,
caratteristica_implementata) AS
    SELECT M.targa, M.disponibile, M.capacità, I.caratteristica
    FROM mezzo M
    LEFT JOIN implementazione I
    ON M.targa = I.targa
    ORDER BY M.targa;
```

### Vista di tutti i dati sui partecipanti

```
CREATE OR REPLACE VIEW dati_partecipanti AS
    SELECT IV.tipo, IV.cf, IV.disponibile, IV.nome, IV.cognome, IV.data_nascita,
    T.numero AS Numero_Di_Telefono, IV.matricola, IV.data_assunzione,
    IV.professione_volontario
    FROM (
        SELECT *
        FROM partecipante P
        LEFT JOIN impiegato I
        USING (cf)
    ) as IV
    LEFT JOIN telefono T
    ON (IV.cf = T.proprietario)
    ORDER BY tipo, cognome;
```

## Trigger

Al fine di fornire reattività al database e di soddisfare tutti i vincoli aziendali specificati, sono stati prodotti dei trigger sulle operazioni di INSERT, UPDATE e DELETE di alcune tabelle.

### INSERT di una campagna

Contestualmente all'inserimento di una campagna è necessario effettuare alcuni controlli sulla validità della richiesta:

- Il responsabile specificato deve essere un impiegato disponibile.
- Non deve essere specificata una data di fine di una campagna per i vincoli ribaditi precedentemente.

```
CREATE OR REPLACE FUNCTION insert_campagna()
RETURNS TRIGGER AS $$
BEGIN
    IF NEW.responsabile NOT IN (
        SELECT I.matricola
        FROM impiegato I
        WHERE I.matricola = NEW.responsabile AND I.cf IN (
            SELECT P.cf
            FROM partecipante P
            WHERE P.tipo = 'impiegato' AND P.disponibile
        )
    ) THEN
        RAISE EXCEPTION 'Il responsabile selezionato non è disponibile';
    END IF;
    IF NEW.data_fine IS NOT NULL THEN
        RAISE EXCEPTION 'Non è possibile creare una campagna con una data di
fine';
    END IF;
    RETURN NEW;
END;
$$
LANGUAGE plpgsql;

CREATE TRIGGER trig_campagna_insert AFTER INSERT ON campagna
FOR EACH ROW
EXECUTE PROCEDURE insert_campagna();
```



## UPDATE di una campagna

È necessario verificare che le nuove informazioni siano adatte ai vincoli specificati in precedenza:

- Il nuovo responsabile deve necessariamente essere un impiegato disponibile.
- Non è possibile concludere una campagna costituita da interventi non ancora conclusi.
- Non è possibile concludere una campagna se non è stato ancora effettuato alcun intervento su alcune delle zone di interesse previste.

```
CREATE OR REPLACE FUNCTION update_campagna()  
RETURNS TRIGGER AS $$  
BEGIN  
    IF NEW.responsabile NOT IN (  
        SELECT I.matricola  
        FROM impiegato I  
        WHERE I.matricola = NEW.responsabile AND I.cf IN (  
            SELECT P.cf  
            FROM partecipante P  
            WHERE P.tipo = 'impiegato' AND P.disponibile  
        )  
    ) THEN  
        RAISE EXCEPTION 'Il responsabile selezionato non è disponibile';  
    END IF;  
  
    IF NEW.data_fine IS NOT NULL THEN  
        IF EXISTS (  
            SELECT *  
            FROM intervento I  
            WHERE I.campagna = NEW.id AND I.data_fine >= LOCALTIMESTAMP  
        ) THEN  
            RAISE EXCEPTION 'Non è possibile concludere una campagna i cui  
interventi non sono ancora conclusi';  
        END IF;  
  
        IF EXISTS (  
            SELECT *  
            FROM interesse I  
            WHERE I.campagna = NEW.id AND I.numero_coperture = 0  
        ) THEN  
            RAISE EXCEPTION 'Non è possibile concludere una campagna le cui zone  
non sono state coperte da almeno un intervento';  
        END IF;  
    END IF;
```

```

        END IF;
        RETURN NEW;
    END;
$$
LANGUAGE plpgsql;

CREATE TRIGGER trig_update_campagna AFTER UPDATE ON campagna
FOR EACH ROW
EXECUTE PROCEDURE update_campagna();

```

## DELETE di una campagna

Contestualmente alla cancellazione di una campagna è necessario verificare che la campagna non sia conclusa (per definizione di storico, consistenza e scelta progettuale).

```

CREATE OR REPLACE FUNCTION delete_campagna()
RETURNS TRIGGER AS $$
    BEGIN
        IF OLD.data_fine IS NOT NULL THEN
            RAISE EXCEPTION 'Non è possibile rimuovere una campagna conclusa';
        END IF;
        RETURN NULL;
    END;
$$
LANGUAGE plpgsql;

CREATE TRIGGER trig_campagna_delete AFTER INSERT ON campagna
FOR EACH ROW
EXECUTE PROCEDURE delete_campagna();

```

## INSERT di un intervento

Sono presenti alcuni controlli ed operazioni da effettuare per garantire il corretto inserimento di un intervento in una campagna:

- La zona d'interesse dell'intervento deve appartenere alle zone previste dalla campagna
- Non è possibile inserire interventi in campagne considerate concluse
- La data d'inizio di un intervento non può essere precedente alla data d'inizio della campagna ad essa associato
- È necessario incrementare il numero di coperture della zona d'interesse della campagna indicata

```

CREATE OR REPLACE FUNCTION insert_intervento()
RETURNS TRIGGER AS $$

```

```

DECLARE nuovo_numero UNSIGNED_INT;
BEGIN
    IF NEW.data_inizio < (
        SELECT C.data_inizio
        FROM campagna C
        WHERE NEW.campagna = C.ID
    ) THEN RAISE EXCEPTION 'Non è possibile aggiungere l'intervento, la sua data
di inizio è antecedente a quella della campagna';
    END IF;

    IF NEW.campagna IN (
        SELECT C.id
        FROM campagna C
        WHERE data_fine IS NOT NULL
    ) THEN RAISE EXCEPTION 'Non è possibile aggiungere l'intervento, la campagna
specificata è terminata';
    END IF;

    IF NEW.zona NOT IN (
        SELECT I.zona
        FROM interesse I
        WHERE I.campagna = NEW.campagna
    ) THEN RAISE EXCEPTION 'La zona specificata non è di interesse per la
campagna selezionata';
    END IF;

    IF NEW.numero IS NULL THEN
        SELECT MAX(numero)+1 INTO nuovo_numero
        FROM intervento
        GROUP BY campagna
        HAVING intervento.campagna = NEW.campagna;
        IF nuovo_numero IS NULL THEN
            nuovo_numero := 1;
        END IF;
        NEW.numero = nuovo_numero;
    END IF;

    UPDATE interesse
    SET numero_coperture = numero_coperture+1
    WHERE zona = NEW.zona AND campagna = NEW.campagna;
    RETURN NEW;
END;

```

```
$$
```

```
LANGUAGE plpgsql;
```

```
CREATE TRIGGER trig_intervento_insert BEFORE INSERT ON intervento
FOR EACH ROW
EXECUTE PROCEDURE insert_intervento();
```

## UPDATE di un intervento

Apportare modifiche ad un intervento è un processo delicato come l'inserimento, è necessario infatti effettuare i seguenti controlli ed operazioni:

- Non è possibile cambiare la campagna di appartenenza.
- La nuova zona indicata deve essere tra le zone d'interesse della campagna
- È necessario aggiornare i numeri di coperture, in caso di modifica sulla zona da coprire

```
CREATE OR REPLACE FUNCTION update_intervento()
RETURNS TRIGGER AS $$
BEGIN
    IF NEW.campagna <> OLD.campagna THEN
        RAISE EXCEPTION 'Non è possibile cambiare la campagna di appartenenza';
    END IF;

    IF NEW.zona <> OLD.zona THEN
        IF NEW.zona NOT IN (
            SELECT I.zona
            FROM interesse I
            WHERE I.campagna = NEW.campagna
        ) THEN
            RAISE EXCEPTION 'La nuova zona specificata non è di interesse per la
campagna selezionata';
        END IF;
        UPDATE interesse
        SET numero_coperture = numero_coperture+1
        WHERE zona = NEW.zona AND campagna = NEW.campagna;
        UPDATE interesse
        SET numero_coperture = numero_coperture-1
        WHERE zona = OLD.zona AND campagna = OLD.campagna;
    END IF;

    RETURN NEW;
END;
$$
LANGUAGE plpgsql;
```

```
CREATE TRIGGER trig_intervento_update AFTER UPDATE ON intervento
FOR EACH ROW
EXECUTE PROCEDURE update_intervento();
```

## DELETE di un intervento

La rimozione dal database di un intervento necessita:

- La verifica che questo non appartenga campagne concluse, aspirando a conservare uno storico attendibile.
- Decrementare il numero di coperture della zona d'interesse.

```
CREATE OR REPLACE FUNCTION delete_intervento()
RETURNS TRIGGER AS $$
BEGIN
    IF OLD.campagna IN (
        SELECT C.id
        FROM campagna C
        WHERE data_fine IS NOT NULL
    ) THEN
        RAISE EXCEPTION 'Non è possibile rimuovere un intervento da una campagna
già terminata';
    END IF;
    UPDATE interesse
    SET numero_coperture = numero_coperture-1
    WHERE zona = OLD.zona AND campagna = OLD.campagna;
    RETURN NULL;
END;
$$
LANGUAGE plpgsql;
```

```
CREATE TRIGGER trig_intervento_delete AFTER DELETE ON intervento
FOR EACH ROW
EXECUTE PROCEDURE delete_intervento();
```

## UPDATE della tabella interesse

La modifica delle zone di una campagna su cui è previsto almeno un intervento richiede:

- La verifica che non si modifichi una zona su cui già è stato effettuato almeno un intervento
- Non venga modificata la campagna o le zone manualmente (corrisponderebbe ad associare ad un'altra campagna le zone d'interesse)

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

```

BEGIN

    IF OLD.numero_coperture>0 AND (NEW.campagna <> OLD.campagna OR NEW.zona <>
OLD.zona) THEN

        RAISE EXCEPTION 'Non è possibile aggiornare la tabella interesse';

    END IF;

    IF NEW.campagna IN (

        SELECT C.id
        FROM campagna C
        WHERE data_fine IS NOT NULL

    ) THEN

        RAISE EXCEPTION 'Non è possibile modificare gli interessi di una
campagna conclusa';

    END IF;

    RETURN NULL;

END;
$$
LANGUAGE plpgsql;

```

```

CREATE TRIGGER trig_interesse_update AFTER UPDATE ON interesse
FOR EACH ROW
EXECUTE PROCEDURE update_interesse();

```

### DELETE dalla tabella interesse

La rimozione di una tupla dalla tabella interesse prevede un controllo sul numero di coperture associato: se questo è maggiore di zero, indica che è stato effettuato almeno un intervento sulla zona specificata

```

CREATE OR REPLACE FUNCTION delete_interesse()
RETURNS TRIGGER AS $$
BEGIN

    IF OLD.numero_coperture > 0 THEN

        RAISE EXCEPTION 'Non è possibile eliminare la zona inserita da quelle
interessate dalla campagna selezionata. Vi è già stato effettuato almeno un
intervento';

    END IF;

    RETURN NULL;

END;
$$
LANGUAGE plpgsql;

```

```
CREATE TRIGGER trig_interesse_delete AFTER DELETE ON interesse
FOR EACH ROW
EXECUTE PROCEDURE delete_interesse();
```

### INSERT di un partecipante

Se il partecipante da inserire è di tipo impiegato, viene effettuato un controllo sulla effettiva presenza nella tabella impiegato dello stesso

```
CREATE OR REPLACE FUNCTION insert_partecipante()
RETURNS TRIGGER AS $$
BEGIN
    IF NEW.tipo = 'impiegato' THEN
        IF NOT EXISTS (
            SELECT *
            FROM impiegato I
            WHERE I.cf = NEW.cf
        ) THEN
            RAISE EXCEPTION 'Impossibile inserire un partecipante di tipo
impiegato che non esiste nella tabella impiegato';
        END IF;
    END IF;
    RETURN NULL;
END;
$$
LANGUAGE plpgsql;
```

```
CREATE TRIGGER trig_partecipante_insert AFTER INSERT ON partecipante
FOR EACH ROW
EXECUTE PROCEDURE insert_partecipante();
```

### UPDATE di un partecipante

Questa è una delle operazioni più delicate, è necessario infatti eseguire i corretti controlli tramite un trigger:

- Nel caso si voglia che un partecipante diventi non disponibile, è necessario verificare che non sia impegnato in interventi futuri
- Nel caso sia un impiegato, è necessario controllare anche che questo non sia il responsabile di una campagna attiva
- Se si vuole rendere impiegato un volontario, è necessario che figuri una tupla con il corrispondente codice fiscale nella tabella impiegato e che l'impiegato non sia responsabile di una o più campagne attive.

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

```

BEGIN
    IF OLD.disponibile = TRUE AND NEW.disponibile = FALSE THEN
        IF NEW.tipo = 'impiegato' AND EXISTS (
            SELECT *
            FROM campagna C
            WHERE C.data_fine IS NULL AND C.responsabile IN (
                SELECT I.matricola
                FROM impiegato I
                WHERE NEW.cf = I.cf
            )
        ) THEN
            RAISE EXCEPTION 'Il partecipante indicato è responsabile di una
campagna attiva';
        END IF;
        IF NEW.disponibile = FALSE AND NEW.cf IN (
            SELECT P.partecipante
            FROM intervento I
            JOIN partecipazione P ON (I.campagna = P.campagna AND i.numero =
P.numero)
            WHERE LOCALTIMESTAMP <= I.data_fine
        ) THEN
            RAISE EXCEPTION 'Non è possibile modificare la disponibilità di un
partecipante coinvolto in un intervento ancora non concluso';
        END IF;
    END IF;

    IF OLD.tipo = 'volontario' AND NEW.tipo = 'impiegato' THEN
        IF NOT EXISTS (
            SELECT *
            FROM impiegato I
            WHERE I.cf = NEW.cf
        ) THEN
            RAISE EXCEPTION 'Impossibile cambiare da volontario a impiegato se
questo non esiste già nella tabella impiegato';
        END IF;
    END IF;

    IF OLD.tipo = 'impiegato' AND NEW.tipo = 'volontario' THEN
        IF EXISTS (
            SELECT *
            FROM campagna
            WHERE responsabile IN (
                SELECT matricola

```



```

        FROM impiegato I
        WHERE I.cf = NEW.cf
    )
) THEN
    RAISE EXCEPTION 'Impossibile cambiare da impiegato a volontario se
questo è stato il responsabile di una campagna';
END IF;
DELETE FROM impiegato WHERE cf = NEW.cf;
END IF;
RETURN NULL;
END;
$$
LANGUAGE plpgsql;

```

```

CREATE TRIGGER trig_partecipante_update AFTER UPDATE ON partecipante
FOR EACH ROW
EXECUTE PROCEDURE update_partecipante();

```

## DELETE impiegato

Se viene rimosso un impiegato dalla tabella impiegato, senza prima renderlo volontario, non ha senso che resti presente nella tabella partecipanti, se il tipo indicato è impiegato.

```

CREATE OR REPLACE FUNCTION delete_impiegato()
RETURNS TRIGGER AS $$
BEGIN
    DELETE FROM partecipante WHERE cf = NEW.cf;
    RETURN NULL;
END;
$$
LANGUAGE plpgsql;

CREATE TRIGGER trig_impiegato_delete AFTER DELETE ON impiegato
FOR EACH ROW
EXECUTE PROCEDURE delete_impiegato();

```

## UPDATE di un mezzo

Se viene richiesta una modifica su un mezzo utilizzato in almeno un intervento ancora non concluso, non è possibile modificarne la disponibilità o la capacità.

```

CREATE OR REPLACE FUNCTION update_mezzo()
RETURNS TRIGGER AS $$
BEGIN
    IF (NOT NEW.disponibile OR NEW.capacità <> OLD.capacità) AND NEW.targa IN (

```

```

        SELECT U.targa
        FROM intervento I
        JOIN utilizzo U ON (I.campagna = U.campagna AND i.numero = U.numero)
        WHERE LOCALTIMESTAMP <= I.data_fine
    ) THEN
        RAISE EXCEPTION 'Non è possibile modificare la disponibilità o la
capacità di un mezzo usato in un intervento ancora non concluso';
    END IF;
    RETURN NULL;
END;
$$
LANGUAGE plpgsql;

CREATE TRIGGER trig_mezzo_update AFTER UPDATE ON mezzo --non posso modificare la
disp di un mezzo se è attivo al momento
FOR EACH ROW
EXECUTE PROCEDURE update_mezzo();

```

## INSERT o UPDATE di partecipazione

È necessario verificare per il corretto utilizzo delle suddette operazioni che:

- L'aggiunta di un partecipante ad interventi appartenenti a campagne terminate non sia permesso
- Non sia permesso impegnare partecipanti non disponibili

```

CREATE OR REPLACE FUNCTION partecipazione_check()
RETURNS TRIGGER AS $$
BEGIN
    IF EXISTS (
        SELECT *
        FROM campagna C
        WHERE C.id = NEW.campagna AND C.data_fine IS NOT NULL
    ) THEN
        RAISE EXCEPTION 'Non è possibile aggiungere partecipanti ad interventi
che appartengono ad una campagna terminata';
    END IF;
    IF EXISTS (
        SELECT *
        FROM partecipante P
        WHERE P.cf = NEW.partecipante AND NOT P.disponibile
    ) THEN
        RAISE EXCEPTION 'Non è possibile impegnare partecipanti non
disponibili';
    END IF;
END;

```

```
        END IF;
        RETURN NEW;
    END;
$$
LANGUAGE plpgsql;

CREATE TRIGGER trig_partecipazione AFTER INSERT OR UPDATE ON partecipazione
FOR EACH ROW
EXECUTE PROCEDURE partecipazione_check();
```