

WeCare

1. Abstract

WeCare è un ente filantropico italiano composto da volontari. Fondato nel 2015 a Padova, a seguito di una richiesta di aiuto da parte di un ente italiano, si occupa di fornire supporto donando i prodotti di cui necessita. Le aziende produttrici, infatti, possono visualizzare i prodotti oggetto di ricerca di WeCare e proporgli dei prezzi di favore. Dapprima operante soltanto in territorio veneto, grazie alla risonanza mediatica del progetto e ai responsi positivi, ha potuto usufruire di ingenti donazioni da parte di persone di spicco e ha ottenuto una crescita esponenziale nelle adesioni. Nel 2020 opera su quasi tutto il territorio italiano.

Uno dei valori fondanti di WeCare è la trasparenza, infatti si impegna a mantenere più chiara possibile la gestione dei fondi, includendo una registrazione delle richieste di supporto soddisfatte. Con l'incremento delle adesioni è aumentata pure la difficoltà nel gestire il capitale regione per regione, viene quindi deciso il passaggio ad un sistema centralizzato garantendo la legittimità e trasparenza delle operazioni. Questo permette, inoltre, un'amministrazione migliore dei fondi.

2. Analisi dei requisiti

2.1 Descrizione testuale dei requisiti

Si vuole realizzare una base di dati che contenga e gestisca le informazioni relative all'associazione WeCare.

WeCare organizza il proprio lavoro in **distretti** ed ogni distretto si occupa di una certa regione italiana. Tuttavia, ci possono essere anche più distretti per regione. Ogni distretto ha un nome e l'indirizzo (via, numero, CAP, città) della sede centrale, dove vengono svolte le riunioni.

Ogni **socio** opera unicamente per un distretto e bisogna tener traccia di alcuni suoi dati, quali codice fiscale, nome, cognome, data di nascita, indirizzo (via, numero, CAP, città), numero di telefono ed email. Tra i soci di ogni distretto viene scelto anche un **presidente**, che deve essere necessariamente in possesso di una laurea in economia, di cui si vuole memorizzare il voto di laurea.

Con cadenza bimensile viene indetta una **riunione** da ogni distretto. Una riunione viene diretta dal presidente del distretto, ma non devono obbligatoriamente partecipare tutti i soci. Una riunione, infatti, può tenersi se si presentano minimo 3 persone, presidente incluso. Lo scopo principale delle riunioni è valutare le richieste in sospeso da parte di enti bisognosi. Un **ente** è caratterizzato dal nome, indirizzo della sede centrale (via, numero, CAP, città), regione in cui risiede, numero di telefono e una eventuale descrizione che ne riassume brevemente il lavoro. Quando un ente ha bisogno di supporto da parte di WeCare, formula una **richiesta** dove specifica i prodotti che necessita in dono e la quantità desiderata. Per ogni **prodotto** dev'essere specificato il nome (e.g. mascherina) ed, eventualmente, una lista delle sue specifiche (e.g. ffp2). La richiesta viene discussa e presa in carico da uno dei distretti che si occupa della regione in cui risiede la sede centrale dell'ente. Si presuppone, quindi, che se un ente è presente nella base di dati è perché ha formulato una richiesta di aiuto.

In una riunione vengono discusse le richieste nuove e/o in sospeso, ossia che non sono ancora state attivate per dare priorità ad altre. Infatti, al termine di ogni incontro, i presenti sono tenuti ad esprimere una votazione anonima mediante una **scheda**, per stabilire quale richiesta attivare per il mese a venire. Purtroppo non sempre tutte le schede sono **valide**, ossia con una preferenza scritta, a volte ci possono essere schede lasciate **bianche** oppure schede con voti **non validi** (una scheda è ritenuta non valida quando sono presenti segni che esulano dalla votazione vera e propria). Quando una richiesta è stata valutata urgente e prioritaria dalla maggioranza relativa, viene **accettata**. Al contrario, le richieste non scelte alla riunione non vengono rifiutate, ma lasciate in

sospeso e rivalutate nei prossimi incontri. Lo scopo è dare una possibilità a tutti, anche se con tempi più dilatati. Ad una richiesta accettata sono associate due date, che corrispondono all'apertura e alla scadenza del bando, infatti, dopo l'approvazione, la richiesta viene resa pubblica, ossia viene pubblicato un bando online con l'elenco degli articoli e relative quantità richieste. Saranno le aziende del territorio a contattare direttamente WeCare per proporre dei preventivi. Tutti i **preventivi**, e le relative aziende, verranno inseriti nella base di dati per poter effettuare un comodo confronto tra tutti. Ogni preventivo deve includere la data di emissione, la richiesta a cui risponde, il totale, l'**azienda** da cui proviene, di cui occorre memorizzare partita IVA, il nome, l'indirizzo della sede centrale (via, numero, CAP, città) e un numero di telefono. Un preventivo è composto da una o più voci: ogni **voce** rappresenta un prodotto della richiesta con il prezzo unitario proposto e la quantità che l'azienda è disposta a fornire a WeCare. È infatti possibile che un'azienda faccia un preventivo incompleto: che non soddisfi pienamente tutte le voci della richiesta (e.g. perché non dispone di tutti gli articoli in magazzino e non ha il tempo di produrli).

Dopo la scadenza del bando, verrà scelto un preventivo **vincitore** tra i proposti e verrà emessa una **fattura** dall'azienda vincitrice, che, dopo aver ricevuto il pagamento da WeCare, provvederà in autonomia a rifornire l'ente richiedente con i materiali concordati. Se non ci saranno stati preventivi o quelli proposti non saranno stati ritenuti convenienti, la richiesta si riterrà conclusa qua.

Per promuovere l'associazione, ogni distretto di WeCare può partecipare a eventi (e.g. fiere) o anche semplicemente allestire banchetti in strada. La partecipazione è di un unico distretto, infatti, essendo l'attività promossa la stessa, sarebbe poco sensato avere più distretti insieme a fare promozione in uno stesso luogo. In queste **occasioni**, di cui si vuole memorizzare la data, l'indirizzo e una breve descrizione, vengono spesso raccolte **donazioni**, che vengono registrate a fine evento come una unica con il totale delle somme ricevute. In concomitanza potrebbero essere necessarie anche delle **spese**, ad esempio in una fiera potrebbe essere richiesto il pagamento di una quota di partecipazione e allestimento.

2.2 Glossario dei termini

Termine	Descrizione	Collegamento
Distretto	L'unità di misura con cui viene gestito il lavoro di WeCare: ogni distretto si occupa di una porzione di territorio di una regione italiana e ci sono regioni con più distretti.	Socio, Presidente, Occasione, Riunione, Transazione
Socio	Volontario membro di WeCare. Prende parte alle attività e alle riunioni di un solo distretto.	Distretto, Riunione
Presidente	Un socio che supervisiona le attività di un distretto. Ce n'è uno per distretto, dirige le riunioni e dev'essere laureato in economia.	Distretto
Riunione	Consultazione con cadenza bimensile tra almeno tre soci di uno stesso distretto dove vengono messe ai voti le richieste di supporto da parte degli enti.	Socio, Distretto, Richiesta, Scheda
Richiesta	Una richiesta di supporto formulata da un ente che verrà votata da un distretto della stessa regione della residenza dell'ente, durante una riunione.	Riunione, Valida, VoceRichiesta, Ente
Accettata	Richiesta che è stata votata dalla maggioranza durante una riunione di un distretto.	Preventivo

Scheda	Mezzo di votazione anonimo usato durante le riunioni dai soci per votare la richiesta da accettare.	Riunione
Preventivo	Emesso da un'azienda, si tratta di un elenco contenente tutti o una parte dei prodotti di una richiesta, le quantità fornite e il prezzo che WeCare dovrà pagare.	Accettata, Azienda, VocePreventivo
Vincitore	Il preventivo scelto da WeCare per soddisfare una specifica richiesta.	Fattura
Occasione	Evento (organizzato da terzi o no) a cui partecipa un distretto per far conoscere WeCare e raccogliere donazioni.	Distretto, Donazione, Spesa
Transazione	Movimento di denaro, in entrata o in uscita, sul conto di WeCare.	Distretto
Donazione	Offerta di denaro ricevuta da un distretto per WeCare durante un'occasione.	Occasione
Spesa	Soldi usati da un distretto per la partecipazione o l'organizzazione di un'occasione.	Occasione
Fattura	Soldi sborsati da WeCare per il pagamento del preventivo vincitore.	Preventivo

2.3 Operazioni tipiche

Dati e stime:

- 20 regioni italiane, 300 distretti totali: una media di 15 distretti per regione;
- una media di 30 soci per distretto → 9000 soci totali;
- una media di 2 voci richiesta per richiesta (idem per il preventivo);
- partecipazione media a riunione: 50%;
- una media di 4 preventivi per richiesta approvata;
- ogni richiesta approvata viene visualizzata da circa una ventina di aziende;
- una media di 2 spese per occasione.

Contesto	Operazione	L/S	Frequenza	
				cadenza
Riunioni	Nuove Riunioni distrettuali	S	20	giorno
Schede	Nuove Schede	S	300	giorno
	Spoglio dei voti (per dichiarare la richiesta approvata)	L	20	giorno
Richieste	Nuove Richieste d'aiuto	S	800	mese
	Nuove VoceRichiesta	S	1600	mese
	Approvate	S	600	mese
	Consultazione delle Approvate da parte delle aziende	L	12000	mese
Preventivi	Nuovi Preventivi	S	2400	mese
	Nuove VocePreventivo	S	4800	mese

	Vincitori (quindi pure Fatture)	S	600	mese
	Consultazioni dei Preventivi per decretare Vincitore	L	600	mese
Occasioni	Nuove occasioni	S	40	giorno
Transazioni	Donazioni	S	40	giorno
	Spese	S	80	giorno
	Fatture	S	20	giorno
	Resoconto entrate/uscite da inizio mese per distretto	L	60	giorno

3. Progettazione concettuale

3.1 Descrizione entità e attributi

(Le chiavi delle entità sono sottolineate e tutti i campi sono NOT NULL tranne quelli specificati)

- **Socio:** rappresenta un socio di un distretto

Chiave primaria: CF

- CF: CHAR(16)
- nome: VARCHAR(50)
- cognome: VARCHAR(50)
- data_nascita: DATE
- email: VARCHAR(100)
- telefono: VARCHAR(15)
- indirizzo: attributo composto
 - via: VARCHAR(50)
 - numero: VARCHAR(3)
 - cap: CHAR(5)
 - città: VARCHAR(20)

L'entità Socio si specializza in una sottocategoria con una generalizzazione parziale:

- **Presidente:** rappresenta il presidente del distretto
 - voto: SMALLINT
 - lode: BOOLEAN

- **Distretto:** rappresenta un distretto dell'associazione WeCare

Chiave primaria: numero + regione

- numero: INT
- regione: ENUM
- nome: VARCHAR(20)
- indirizzo: attributo composto
 - via: VARCHAR(50)
 - numero: VARCHAR(3)
 - cap: CHAR(5)
 - città: VARCHAR(20)

- **Ente:** rappresenta un ente che richiede assistenza a WeCare

Chiave primaria: nome + regione

- nome: VARCHAR(50)
- regione: ENUM
- indirizzo: attributo composto

- via: VARCHAR(50)
 - numero: VARCHAR(3)
 - cap: CHAR(5)
 - città: VARCHAR(20)
- telefono: VARCHAR(15)
- descrizione: VARCHAR(100) [può essere NULL]
- **Azienda:** rappresenta un'azienda che fa un preventivo per soddisfare una richiesta
 - Chiave primaria:** PIVA
 - PIVA: CHAR(11)
 - nome: VARCHAR(50)
 - telefono: VARCHAR(15)
 - indirizzo: attributo composto
 - via: VARCHAR(50)
 - numero: VARCHAR(3)
 - cap: CHAR(5)
 - città: VARCHAR(20)
- **Occasione:** rappresenta un'occasione sfruttata da un distretto per promuovere WeCare
 - Chiave primaria:** data + indirizzo
 - data: DATE
 - indirizzo: attributo composto
 - via: VARCHAR(50)
 - numero: VARCHAR(3)
 - cap: CHAR(5)
 - città: VARCHAR(20)
 - descrizione: VARCHAR(255)
- **Prodotto:** rappresenta un prodotto che è stato richiesto da un ente
 - Chiave primaria:** id
 - id: SERIAL
 - nome: VARCHAR(50)
 - specifiche: VARCHAR(150) [può essere NULL]
- **Richiesta:** rappresenta la richiesta di supporto fatta da un ente a WeCare
 - Chiave primaria:** ente + creazione
 - creazione: DATE

L'entità Richiesta si specializza in una sottocategoria con generalizzazione parziale:

 - **Accettata:** rappresenta una richiesta che è stata accettata e per cui adesso possono essere inoltrati preventivi dalle aziende
 - apertura: DATE
 - chiusura: DATE
- **VoceRichiesta:** rappresenta i prodotti con relative quantità che fanno parte di una richiesta
 - Chiave primaria:** richiesta + prodotto
 - quantità: INT
- **Riunione:** rappresenta una riunione del distretto dove viene discussa e votata una o più richieste d'aiuto in sospeso fatte dagli enti
 - Chiave primaria:** distretto + data
 - data: DATE

- **Scheda:** rappresenta la scheda usata anonimamente da un socio per esprimere la propria preferenza sulla richiesta da soddisfare
Chiave primaria: numero
 - numero: SERIAL
 L'entità Scheda si specializza in tre sottocategorie con generalizzazione totale:
 - **Nulla:** rappresenta una scheda con voto non valido
 - **Bianca:** rappresenta una scheda senza una preferenza
 - **Valida:** rappresenta una scheda con preferenza verso una richiesta
- **Preventivo:** rappresenta un preventivo fatto da un'azienda per soddisfare una richiesta
Chiave primaria: azienda + accettata
 - emissione: DATE
 - totale: DECIMAL(8,2)
 L'entità Preventivo si specializza in una sottocategoria con generalizzazione parziale:
 - **Vincitore:** rappresenta il preventivo scelto dall'associazione WeCare
 - dichiarazione: DATE
- **VocePreventivo:** rappresenta una voce all'interno del preventivo fatto da un'azienda
Chiave primaria: voce richiesta + preventivo
 - quantita: INT
 - prezzo: DECIMAL(8,2)
- **Transazione:** rappresenta una transazione monetaria effettuata sul saldo comune dell'associazione WeCare
Chiave primaria: time + distretto
 - time: TIMESTAMP
 - importo: DECIMAL(8,2)
 L'entità Transazione si specializza in tre sottocategorie con generalizzazione totale:
 - **Donazione:** rappresenta l'insieme delle donazioni fatte a favore di un distretto in un'occasione
 - **Spesa:** rappresenta la spesa di un distretto
 - giustificazione: VARCHAR(255)
 - **Fattura:** rappresenta l'acquisto del preventivo vincitore
 - progressivo: SERIAL

Altri vincoli non rappresentabili nello schema concettuale

Oltre ai vincoli rappresentati nella descrizione delle entità e delle relazioni sono presenti anche i seguenti:

- La lode nel voto di laurea del presidente può essere aggiunta se e solo se voto=110;
- La richiesta per essere accettata deve ottenere la maggioranza relativa dei voti.

3.2 Descrizione relazioni

Socio - Distretto: iscrizione

- Un socio è iscritto a un solo distretto. (1,1)
- Un distretto ha come iscritti uno o più soci. (1,N)

Presidente - Distretto: presidenza

- Un presidente presiede un solo distretto. (1,1)
- Un distretto ha un solo presidente. (1,1)

Socio - Riunione: presenza

- Un socio può non aver presenziato a nessuna riunione o aver presenziato a una o più. (0,N)
- A una riunione presenziano almeno tre soci. (3,N)

Distretto - Riunione: organizza

- Un distretto può non aver ancora organizzato riunioni o averne organizzata una o più. (0,N)
- Una riunione è organizzata da un solo distretto. (1,1)

Riunione - Richiesta: valutazione

- Ad una riunione vengono valutate una o più richieste. (1,N)
- Una richiesta può non essere stata ancora valutata in una riunione o essere stata valutata in una o più riunioni. (0,N)

Richiesta - Ente: formulazione

- Una richiesta viene formulata da un solo ente. (1,1)
- Un ente può aver formulato una o più richieste. (1,N)

Scheda - Riunione: votazione

- Una scheda viene usata per votare ad una riunione. (1,1)
- In una riunione vengono usate minimo tre schede per votare. (3,N)

Valida - Richiesta: preferenza

- Una scheda valida esprime la preferenza per una richiesta sola. (1,1)
- Una richiesta riceve nessuno oppure uno o più voti da schede valide. (0,N)

Richiesta - VoceRichiesta: composizione

- Una richiesta è composta una o più voci. (1,N)
- Una voce richiesta fa parte di una sola richiesta. (1,1)

VoceRichiesta - Prodotto: consiste

- Una voce richiesta consiste di un solo prodotto. (1,1)
- Un prodotto è presente in una o più voci di richiesta. (1,N)

Preventivo - Azienda: emissione

- Un preventivo viene emesso da un'azienda. (1,1)
- Un'azienda ha emesso uno o più preventivi. (1,N)

Preventivo - Accettata: proposta

- Un preventivo viene proposto per una proposta accettata. (1,1)
- Una proposta accettata ha zero o più proposte di preventivi. (0,N)

VoceRichiesta - VocePreventivo: soddisfa

- Una voce richiesta può non ricevere nessun preventivo quindi non riguardare alcuna voce preventivo, oppure riguardarne una o più. (0,N)
- Una voce preventivo riguarda una sola voce richiesta. (1,1)

Preventivo - VocePreventivo: composizione

- Un preventivo è composto da almeno una voce preventivo. (1,N)
- Una voce preventivo riguarda un solo preventivo. (1,1)

Vincitore - Fattura: acquisto

- La fattura per il preventivo vincitore è una sola ed è possibile che non sia stata ancora emessa. (0,1)
- Una fattura riguarda l'acquisto di un solo preventivo vincitore. (1,1)

Occasione - Spesa: necessita

- La partecipazione ad un'occasione potrebbe non necessitare di spese oppure richiederne una o più. (0,N)
- Una spesa è necessaria per partecipare ad una sola specifica occasione. (1,1)

Occasione - Donazione: presso

- Ad un'occasione si possono non ricevere donazioni oppure riceverne una o più, che in entrambi i casi viene registrata come una unica. (0,1)
- Una donazione è stata ricevuta presso un'occasione specifica. (1,1)

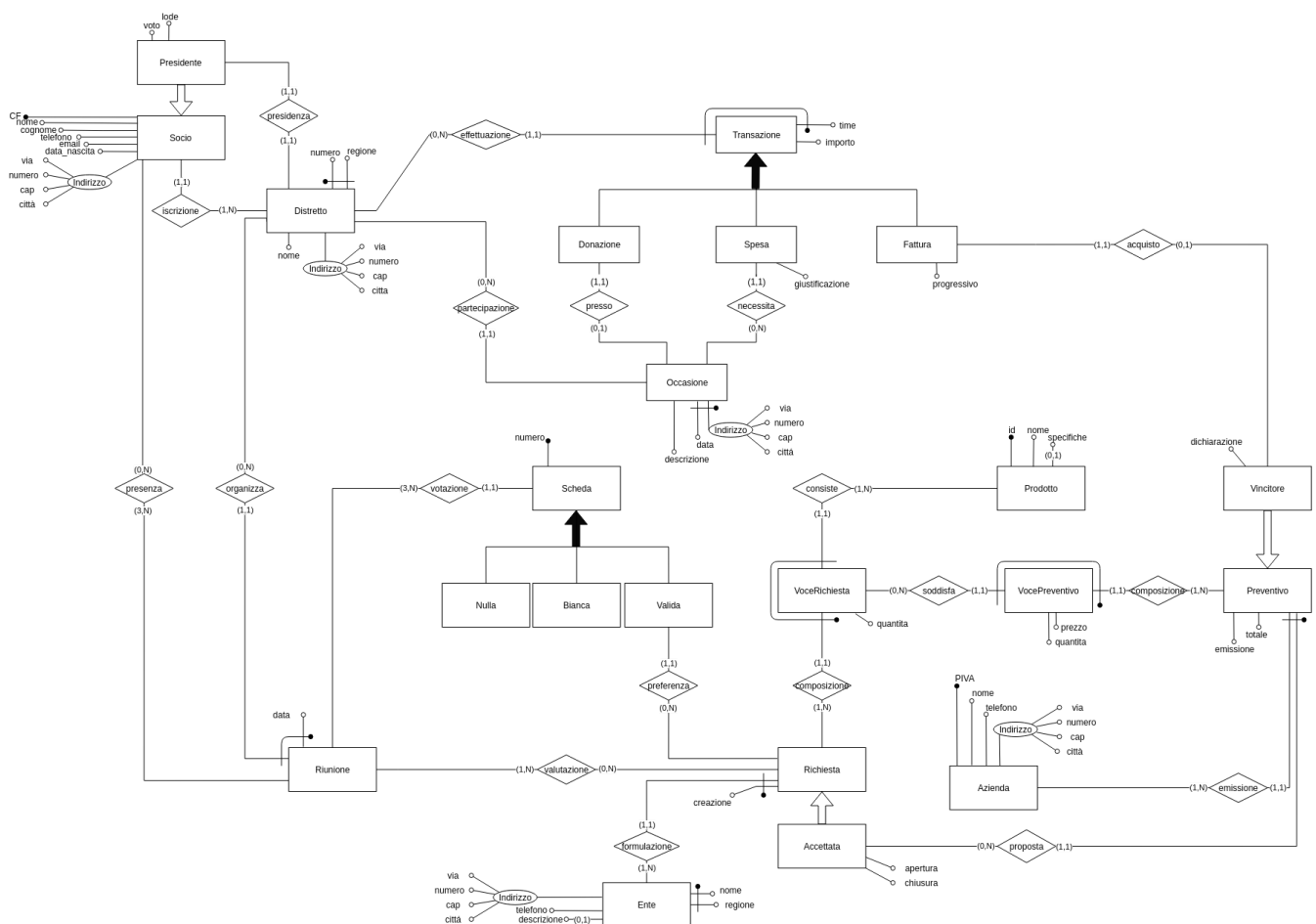
Distretto - Occasione: partecipazione

- Un distretto può aver partecipato a nessuna, a una o a più occasioni. (0,N)
- Ad un'occasione partecipa un solo distretto. (1,1)

Distretto - Transazione: effettuazione

- Un distretto può non aver effettuato transazioni o averne effettuata una o più di una. (0,N)
- Una transazione è effettuata da un solo distretto. (1,1)

3.3 Schema concettuale (E-R) ([link a .svg](#))



4. Progettazione logica

4.1 Ristrutturazione dello schema: 4.1.1 Analisi delle ridondanze

Presidenza

Anche se la relazione *presidenza* è ridondante (un socio può diventare presidente solo del distretto a cui è iscritto), viene mantenuta per imporre il vincolo che un distretto deve avere un unico presidente e perché Presidente ha due campi propri, non presenti in Socio o Distretto.

Totale in Preventivo

Il *totale* in Preventivo è ridondante, infatti può essere calcolato sommando il risultato di $\text{prezzo} \times \text{quantità}$ (attributi in VocePreventivo) per tutte le istanze relative al preventivo specifico. Le operazioni principali che lo coinvolgono e determinano il carico applicativo sono:

Operazione 1: Creazione di una nuova voce nel preventivo. Ogni aggiunta di una voce al preventivo implica l'incremento di *totale*.

Operazione 2: Ricerca preventivi relativi alla richiesta in analisi per scelta del Vincitore.

Concetto	E/R	Volume
Preventivo	E	144.000
Voce Preventivo	E	288.000
Composizione	R	288.000

# op.	L/S	Frequenza
1	S	160 al giorno
2	L	20 al giorno

Operazione 1

Concetto	E/R	Tipo	Accessi (no ridondanza)	Accessi (con ridondanza)
Voce Preventivo	E	S	1	1
Composizione	R	S	1	1
Preventivo	E	L	0	1
Preventivo	E	S	0	1
TOTALE SINGOLO			4 (la scrittura vale doppio)	7 (la scrittura vale doppio)
TOTALE GIORNALIERO			640	1120

Operazione 2

Concetto	E/R	Tipo	Accessi (no ridondanza)	Accessi (con ridondanza)
Preventivo	E	L	4	4
Voce Preventivo	E	L	8	0
Composizione	R	L	8	0
TOTALE SINGOLO			20	4
TOTALE GIORNALIERO			400	80

Costo totale giornaliero senza ridondanza: 1040 accessi

Costo totale giornaliero con ridondanza: 1200 accessi

L'attributo *totale* è di tipo DECIMAL(8,2), dimensione 4 Bytes, e occupa $4 \times 144.000 = 576\text{KB}$.

Visto che la presenza del dato ridondante causa un incremento dell'occupazione di memoria pari a 576KB e un incremento nel numero di accessi giornalieri pari a 160 (1200-1040), è stato deciso di rimuovere la ridondanza.

4.1.2 Eliminazione delle generalizzazioni

Socio → Presidente:

Essendo una generalizzazione non totale, anche se il presidente perdesse il ruolo rimarrebbe comunque sempre un socio, pertanto è giusto mantenere le due entità distinte e in relazione tra loro. Inoltre, unire Presidente e Socio in un'unica entità portando in Socio gli attributi di Presidente e permettendo cardinalità nulla in questi, non converrebbe poiché, essendo il numero dei presidenti nettamente inferiore rispetto ai Soci, si avrebbe un alto spreco di memoria.

Scheda:

Si può eliminare la generalizzazione aggiungendo l'attributo *tipologia* a Scheda, che identifica una scheda come "nulla", "bianca" o "valida". Se *tipologia*="valida", allora la scheda è in relazione con la preferenza, altrimenti no. Per questo motivo, Scheda ha cardinalità (0,1) nella relazione *preferenza* con Richiesta.

Richiesta:

Accettata rispetto a Richiesta ha in più soltanto due attributi di tipo DATE: apertura e chiusura del bando. Un attributo DATE occupa 3 bytes, quindi per ogni richiesta non ancora accettata ci sono 6 bytes sprecati con valori null. Tuttavia, considerando che tutte le richieste prima o poi verranno approvate e che quelle non ancora approvate sono meno delle approvate (mediamente $\frac{1}{4}$ del totale), l'accorpamento delle due entità in un'unica entità Richiesta risulta essere la scelta migliore.

Transazione:

Una soluzione ibrida risulta essere la soluzione migliore. L'aggiunta dell'attributo *tipologia* permette di distinguere "donazione" da "spesa" e da "fattura" e di accorpare Donazione all'entità padre Transazione. Al contrario, Spesa e Fattura restano in entità distinte perché Spesa porterebbe in Transazione altri 2 attributi e Fattura ben 3, ottenendo quindi 5 attributi null per ogni donazione. Un altro motivo è la necessità da parte di WeCare, per questioni di contabilità, di avere un accesso separato alle Spese e alle Fatture.

Preventivo → Vincitore:

La scelta è di dividere le due entità perché analizzando le statistiche sui preventivi, è emerso che circa $\frac{1}{4}$ dei preventivi proposti diventano vincitori, quindi accorpando gli attributi in un'unica entità si avrebbe troppo spreco di memoria.

4.1.3 Partizionamento/accorpamento di entità

Indirizzo:

L'attributo composto *indirizzo* diventa un unico attributo di tipo VARCHAR contenente via, numero, cap e città. La scelta si basa sul fatto che è un'informazione interrogata di rado, e, soprattutto, non si riscontrano vantaggi nel renderlo un'entità a sé o nel gestire i singoli attributi separatamente, in quanto quando viene richiesto un indirizzo lo si vuole nella sua interezza.

4.1.4 Scelta degli identificatori primari

(PK=Primary Key | FK=Foreign Key)

Richiesta:

La PK era composta da *creazione* ed *ente*, con quest'ultimo formato da *nome* e *regione*. Siccome Richiesta ha molteplici riferimenti dove diventa FK (nella relazione valutazione, nelle entità VoceRichiesta, Preventivo e Scheda), sarebbe stato pesante portarsi dietro 3 attributi. Viene quindi deciso di aggiungere un campo id che faccia da PK.

Riunione:

Avere come PK *data* e *distretto*, con quest'ultimo formato da *numero* e *regione*, diventerebbe pesante, soprattutto nella tabella di congiunzione "*presenza*". Inoltre in "*presenza*" (che rappresenta i soci presenti ad una riunione) si creerebbe una ridondanza perché, dato che un socio può partecipare solo a riunioni del suo distretto, è inutile ribadire il distretto dove viene fatta la riunione. Viene quindi deciso di aggiungere un id.

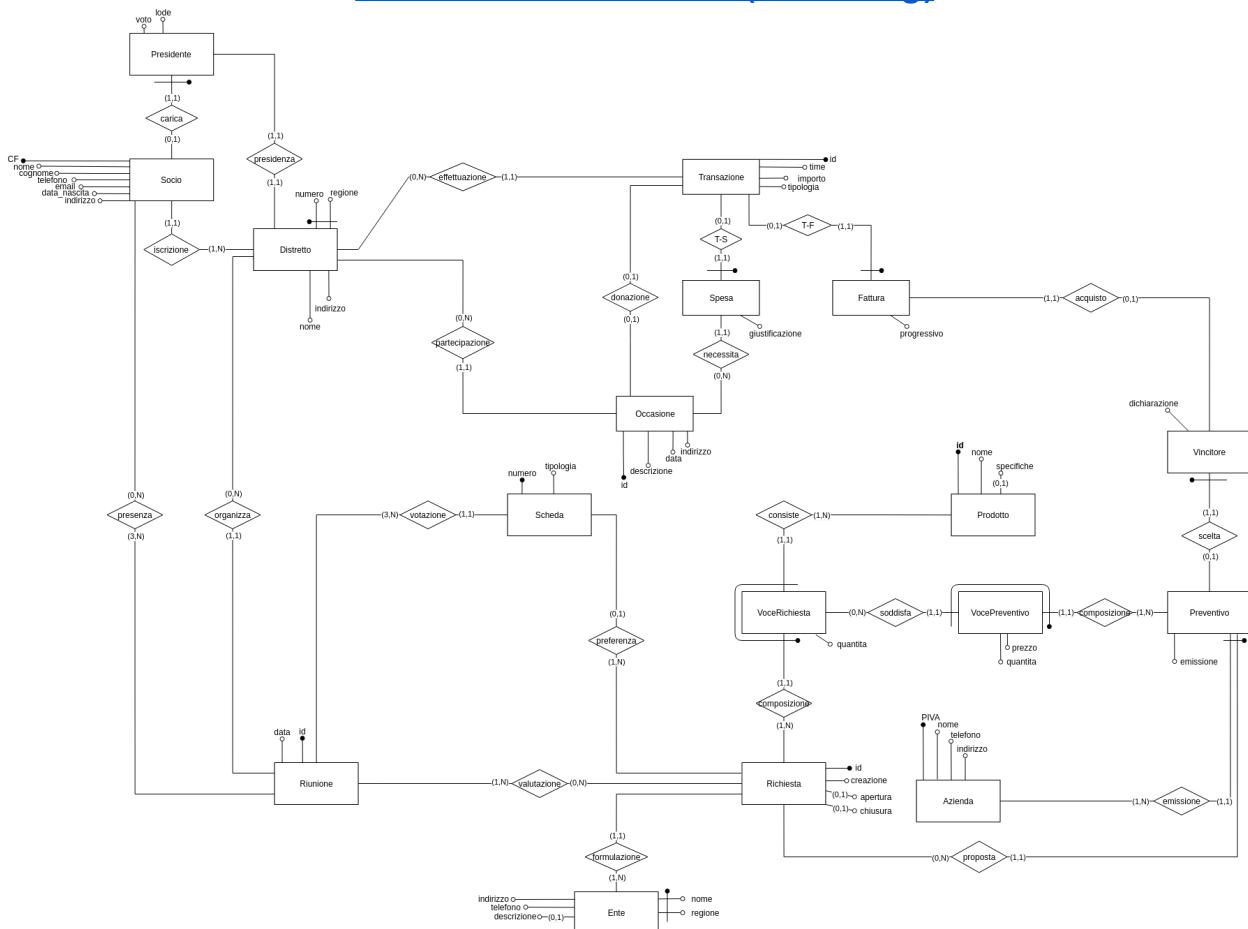
Occasione:

La PK era composta da *data* e *indirizzo*. Questa viene poi utilizzata come FK in Transazione ogni volta che la tipologia è "*donazione*". Data la mole elevata di dati prevista per le transazioni diventa un grande spreco di memoria avere due riferimenti al posto di uno solo. Viene quindi deciso di aggiungere un id.

Transazione:

La PK era composta da *time* e *distretto* con quest'ultimo formato da *numero* e *regione*, essendo riferito in Spesa e Fattura, ogni riferimento comporta l'utilizzo di 3 campi. Risulta conveniente quindi scegliere un id per Transazione.

4.2 Schema relazionale ([link a .svg](#))



5. Implementazione dello schema logico

Socio (<u>cf</u> , nome, cognome, telefono, email, data_nascita, indirizzo, n_distretto, regione)	Socio.n_distretto → Distretto.numero Socio.regione → Distretto.regione
Presidente (<u>cf</u> , voto, lode)	Presidente.cf → Socio.cf
Distretto (<u>regione</u> , <u>numero</u> , presidente, nome, indirizzo)	Distretto.presidente → Presidente.cf
Riunione (<u>id</u> , regione, n_distretto, data)	Riunione.regione → Distretto.regione Riunione.n_distretto → Distretto.numero
Presenza (<u>socio</u> , <u>riunione</u>)	Presenza.socio → Socio.cf Presenza.riunione → Riunione.id
Ente (<u>nome</u> , <u>regione</u> , indirizzo, telefono, descrizione)	
Richiesta (<u>id</u> , ente, regione, creazione, apertura, chiusura)	Richiesta.ente → Ente.nome Richiesta.regione → Ente.regione
Valutazione (<u>riunione</u> , <u>richiesta</u>)	Valutazione.riunione → Riunione.id Valutazione.richiesta → Richiesta.id
Prodotto (<u>id</u> , nome, specifiche)	
VoceRichiesta (<u>prodotto</u> , <u>richiesta</u> , quantita)	VoceRichiesta.prodotto → Prodotto.id VoceRichiesta.richiesta → Richiesta.id
Azienda (<u>piva</u> , nome, telefono, indirizzo)	
Preventivo (<u>piva</u> , <u>richiesta</u> , emissione)	Preventivo.piva → Azienda.piva Preventivo.richiesta → Richiesta.id
VocePreventivo (<u>azienda</u> , <u>prodotto</u> , <u>richiesta</u> , quantita, prezzo)	VocePreventivo.azienda → Preventivo.piva VocePreventivo.richiesta → Preventivo.richiesta VocePreventivo.prodotto → VoceRichiesta.prodotto VocePreventivo.richiesta → VoceRichiesta.richiesta
Scheda (<u>numero</u> , riunione, tipologia, preferenza)	Scheda.riunione → Riunione.id Scheda.preferenza → Richiesta.id
Vincitore (<u>piva</u> , <u>richiesta</u> , dichiarazione)	Vincitore.piva → Preventivo.piva Vincitore.richiesta → Preventivo.richiesta
Occasione (<u>id</u> , indirizzo, data, descrizione, regione, n_distretto)	Occasione.regione → Distretto.regione Occasione.n_distretto → Distretto.numero
Transazione (<u>id</u> , time, regione, n_distretto, importo, tipologia, donato_presso)	Transazione.regione → Distretto.regione Transazione.n_distretto → Distretto.numero Transazione.donato_presso → Occasione.id
Spesa (<u>transazione</u> , giustificazione, occasione)	Spesa.transazione → Transazione.id Spesa.occasione → Occasione.id
Fattura (<u>transazione</u> , progressivo, piva, richiesta)	Fattura.transazione → Transazione.id Fattura.piva → Vincitore.piva Fattura.richiesta → Vincitore.richiesta

6. Query e indici

1. Per ogni distretto mostrare il totale delle donazioni che ha ottenuto, il totale delle spese effettuate, il totale delle fatture emesse e il ricavato (donazioni-(fatture+spese)):

```
DROP VIEW IF EXISTS totaleFatt; DROP VIEW IF EXISTS totaleDonaz; DROP VIEW IF EXISTS totaleSpese;
CREATE VIEW totaleFatt AS (SELECT Distretto.nome, SUM(importo) AS TOT_FATTURE
FROM Distretto JOIN Transazione ON Distretto.numero = Transazione.n_distretto AND Distretto.regione = Transazione.regione
WHERE Transazione.tipologia = 'FATTURA'
GROUP BY Distretto.nome);

CREATE VIEW totaleDonaz AS (SELECT Distretto.nome, SUM(importo) AS TOT_DONAZIONI
FROM Distretto JOIN Transazione ON Distretto.numero = Transazione.n_distretto AND Distretto.regione = Transazione.regione
WHERE Transazione.tipologia = 'DONAZIONE'
GROUP BY Distretto.nome);

CREATE VIEW totaleSpese AS (SELECT Distretto.nome, SUM(importo) AS TOT_SPESE
FROM Distretto JOIN Transazione ON Distretto.numero = Transazione.n_distretto AND Distretto.regione = Transazione.regione
WHERE Transazione.tipologia = 'SPESA'
GROUP BY Distretto.nome);

SELECT DISTINCT totaleFatt.nome, TOT_DONAZIONI, TOT_SPESE, TOT_FATTURE, (TOT_DONAZIONI-(TOT_SPESE+TOT_FATTURE)) AS NETTO
FROM totaleFatt JOIN totaleDonaz ON totaleFatt.nome=totaleDonaz.nome JOIN totaleSpese ON totaleDonaz.nome=totaleSpese.nome;
```

	nome character varying (20)	tot_donazioni numeric	tot_spese numeric	tot_fatture numeric	netto numeric
1	Padova1	543.92	179.00	18967.00	-18602.08
2	Padova2	341.00	160.00	11946.00	-11765.00
3	Roma1	329.00	130.00	2253.00	-2054.00
4	Venezia1	404.10	253.00	9190.00	-9038.90

2. Elencare in ordine crescente i distretti che hanno approvato più richieste contenenti il prodotto “Mascherina” nel periodo che va dal 15.10.2020 al 16.11.2020:

```
SELECT Riunione.regione, Riunione.n_distretto, COUNT(*) AS N_richiesteApprovate
FROM Riunione JOIN Valutazione ON Riunione.id = Valutazione.riunione JOIN Richiesta ON Valutazione.richiesta = Richiesta.id
JOIN VoceRichiesta ON Richiesta.id = VoceRichiesta.richiesta JOIN Prodotto ON VoceRichiesta.prodotto = Prodotto.id
WHERE Prodotto.nome='Mascherina' AND Richiesta.apertura >='2020.10.15' AND Richiesta.apertura <='2020.11.16'
GROUP BY (Riunione.regione, Riunione.n_distretto)
ORDER BY N_richiesteApprovate DESC;
```

	regione regioni	n_distretto integer	n_richiesteapprovate bigint
1	LAZIO	1	2
2	VENETO	3	1

3. Elencare in ordine crescente le tre aziende che hanno ricevuto più soldi da parte di WeCare e, per ognuna, il numero di preventivi vincitori che detiene:

```
SELECT T2.PIVA, T1.Azienda, T1.N_Vincite, T2.Soldi_ricevuti
FROM
(SELECT Azienda.piva AS PIVA, Azienda.nome AS Azienda, count(*) AS N_Vincite
FROM Azienda INNER JOIN Preventivo ON Azienda.piva = Preventivo.piva INNER JOIN Vincitore ON Preventivo.piva = Vincitore.piva
GROUP BY Azienda.piva) AS T1
JOIN
(SELECT Vincitore.piva AS PIVA, SUM(Transazione.importo) AS Soldi_ricevuti
FROM Vincitore INNER JOIN Fattura ON Vincitore.piva = Fattura.piva INNER JOIN Transazione ON Fattura.transazione = Transazione.id
GROUP BY Vincitore.piva) AS T2
ON T2.PIVA = T1.PIVA
ORDER BY Soldi_ricevuti DESC
LIMIT 3;
```

	piva character (11)	azienda character varying (50)	n_vincite bigint	soldi_ricevuti numeric
1	9416760333	Healthcare s.a.s.	4	20020.00
2	8238401722	FarmaPh s.r.l.	1	10210.00
3	7607418259	Grandi Magazzini Ferrarin s.r.l.	1	8007.00

4. Elencare per ciascuna regione il tempo (in giorni) minimo, medio e massimo impiegato per portare a termine le richieste, il numero di richieste portate a termine e quelle ancora aperte:

```

DROP VIEW IF EXISTS RichiesteAperte;
CREATE VIEW RichiesteAperte AS (
    SELECT id, ente, regione, creazione, apertura, chiusura FROM Richiesta WHERE apertura IS NOT NULL AND chiusura IS NULL);
DROP VIEW IF EXISTS RichiestaConclusaTempoImpiegato;
CREATE VIEW RichiestaConclusaTempoImpiegato AS (
    SELECT id, ente, regione, EXTRACT(EPOCH FROM (chiusura::timestamp - apertura::timestamp))/86400::int AS tempo FROM Richiesta
    WHERE apertura IS NOT NULL AND chiusura IS NOT NULL);

SELECT rcti.regione, MIN(rcti.tempo) AS minimo, MAX(rcti.tempo) AS massimo, AVG(rcti.tempo) AS medio,
    COUNT(rcti.id) AS concludere, COUNT(ra.id) AS aperte
FROM RichiestaConclusaTempoImpiegato AS rcti
FULL OUTER JOIN RichiesteAperte AS ra
ON rcti.id = ra.id GROUP BY (rcti.regione);

```

	regione regioni	minimo double precision	massimo double precision	medio double precision	concludere bigint	aperte bigint
1	LAZIO	30	31	30.333333333333332	3	0
2	VENETO	30	31	30.333333333333332	9	0

5. Per ogni richiesta (approvata o meno) trovare il distretto che l'ha approvata e determinare durante quale riunione e con quanti voti è passata:

```

DROP VIEW IF EXISTS VotiPerRichiesta CASCADE;
CREATE VIEW VotiPerRichiesta AS (
    SELECT riunione, preferenza, count(numero) AS num_voti FROM Scheda
    WHERE preferenza IS NOT NULL GROUP BY (preferenza, riunione) ORDER BY riunione ASC
);

DROP VIEW IF EXISTS VincitorePerRiunione;
CREATE VIEW VincitorePerRiunione AS (
    SELECT * FROM VotiPerRichiesta a
    WHERE num_voti = (SELECT MAX(num_voti) FROM VotiPerRichiesta WHERE riunione = a.riunione)
);

SELECT ric.id, riu.n_distretto, riu.regione, v.riunione, v.num_voti FROM Riunione AS riu FULL JOIN VincitorePerRiunione AS v
ON riu.id = v.riunione
FULL JOIN Richiesta AS ric ON v.preferenza = ric.id
ORDER BY ric.id ASC;

```

	id integer	n_distretto integer	regione regioni	riunione integer	num_voti bigint
1	1	1	VENETO	1	3
2	2	2	VENETO	2	3
3	3	3	VENETO	3	4
4	4	1	LAZIO	4	2

(il risultato è stato riportato parzialmente per questione di spazio)

6: Per ciascun DISTRETTO all'interno della regione "VENETO" ricavare quante presenze ha totalizzato dall'inizio dell'anno "2020" (la somma di ogni "presenza") e inoltre uno tra i soci meno presenti con il suo numero di presenze e lo stesso per uno tra i più presenti:

```
DROP VIEW IF EXISTS PresenzePerDistretto;
CREATE VIEW PresenzePerDistretto AS (
    SELECT s.n_distretto, s.regione, COUNT(p.socio) AS presenze, EXTRACT(YEAR FROM r.data) AS anno FROM Presenza AS p
    INNER JOIN Socio AS s ON p.socio = s.cf INNER JOIN Riunione AS r ON p.riunione = r.id GROUP BY (s.n_distretto, s.regione, anno)
);

DROP VIEW IF EXISTS PresenzaPerSocio;
CREATE VIEW PresenzaPerSocio AS (
    SELECT p.socio, n_distretto, regione, COUNT(p.socio) AS num_presenze FROM Presenza AS p INNER JOIN Socio AS s ON s.cf = p.socio
    GROUP BY (socio, n_distretto, regione) ORDER BY num_presenze
);

SELECT ppd.regione, ppd.n_distretto, ppd.presenze, ppd.anno, pp.socio AS "piu presente", pp.num_presenze AS "presenze massime", mp.socio AS "meno presente",
mp.num_presenze AS "presenze minime"
FROM PresenzePerDistretto AS ppd
FULL JOIN (
    SELECT DISTINCT ON (n_distretto, regione) * FROM PresenzaPerSocio AS p1
    WHERE num_presenze = (SELECT MAX(num_presenze) FROM PresenzaPerSocio AS p2 WHERE p1.regione = p2.regione AND p1.n_distretto = p2.n_distretto LIMIT 1)
) AS pp ON ppd.n_distretto = pp.n_distretto AND ppd.regione = pp.regione
FULL JOIN (
    SELECT DISTINCT ON (n_distretto, regione) * FROM PresenzaPerSocio AS p1
    WHERE num_presenze = (SELECT MIN(num_presenze) FROM PresenzaPerSocio AS p2 WHERE p1.regione = p2.regione AND p1.n_distretto = p2.n_distretto LIMIT 1)
) AS mp ON ppd.n_distretto = mp.n_distretto AND ppd.regione = mp.regione
WHERE anno = 2020 AND ppd.regione = 'VENETO' ORDER BY (ppd.regione, ppd.n_distretto);
```

	regione	n_distretto	presenze	anno	piu presente	presenze massime	meno presente	presenze minime
	regioni	integer	bigint	double precision	character (16)	bigint	character (16)	bigint
1	VENETO	1	11	2020	RSSMRA70S02G224K	3	LFACNA59U13P130U	1
2	VENETO	2	12	2020	CHRSTA00P02A224K	3	KRASMI90E71V291P	1
3	VENETO	3	13	2020	ANDTRO69B42E700J	3	LGUBTE82E50H905W	1

Indice

Più aumenta la notorietà di WeCare e maggiore sarà il numero delle richieste nuove ricevute giornalmente, tuttavia queste riguarderanno per la maggior parte prodotti già esistenti nella tabella Prodotti, dato che questa conterrà già una grande varietà di articoli, quindi Prodotti non sarà una tabella che riceverà molti inserimenti. Inoltre, Prodotto viene consultata spesso, infatti è interrogata durante la valutazione di una richiesta, nella creazione di un preventivo e nella scelta del preventivo vincitore, pertanto è utile indicizzare la tabella:

```
CREATE INDEX idx_idProdotto ON Prodotto(id);
```

7. Script.cpp

Perché lo script funzioni è necessario inserire in PG_HOST l'host, in PG_USER il nome utente, in PG_DB il nome del database, in PG_PASS la password per accedervi e in PG_PORT il numero di porta. Inoltre bisogna includere nel codice il file libpq-fe.h.

Per la compilazione dello script bisogna specificare il percorso dell'include, preceduto dalla stringa "-I", dove il compilatore cercherà il file "libpq-fe.h"; inoltre è necessario specificare la libreria libpq tramite "-lpq":

```
g++ script.cpp -o exe -I /usr/include/postgresql -lpq
```

È stato deciso di creare un unico file per tutte le query per due motivi principali: il primo è poter fornire una semplice interfaccia da cui selezionare quale query eseguire tra le sei, così da agevolare utenti più inesperti e il secondo è che, per evitare ripetizione di codice uguale, è stata creata una funzione unica e generica in grado di formattare tabularmente l'output di una query qualsiasi specificando il numero di tuple e il numero di campi.

Nello script la seconda query è riportata in versione parametrica: è richiesto all'utente l'inserimento di tre parametri: il nome del prodotto da cercare e le due date che circoscrivono il periodo su cui si vuole effettuare la ricerca.