

Schema logico

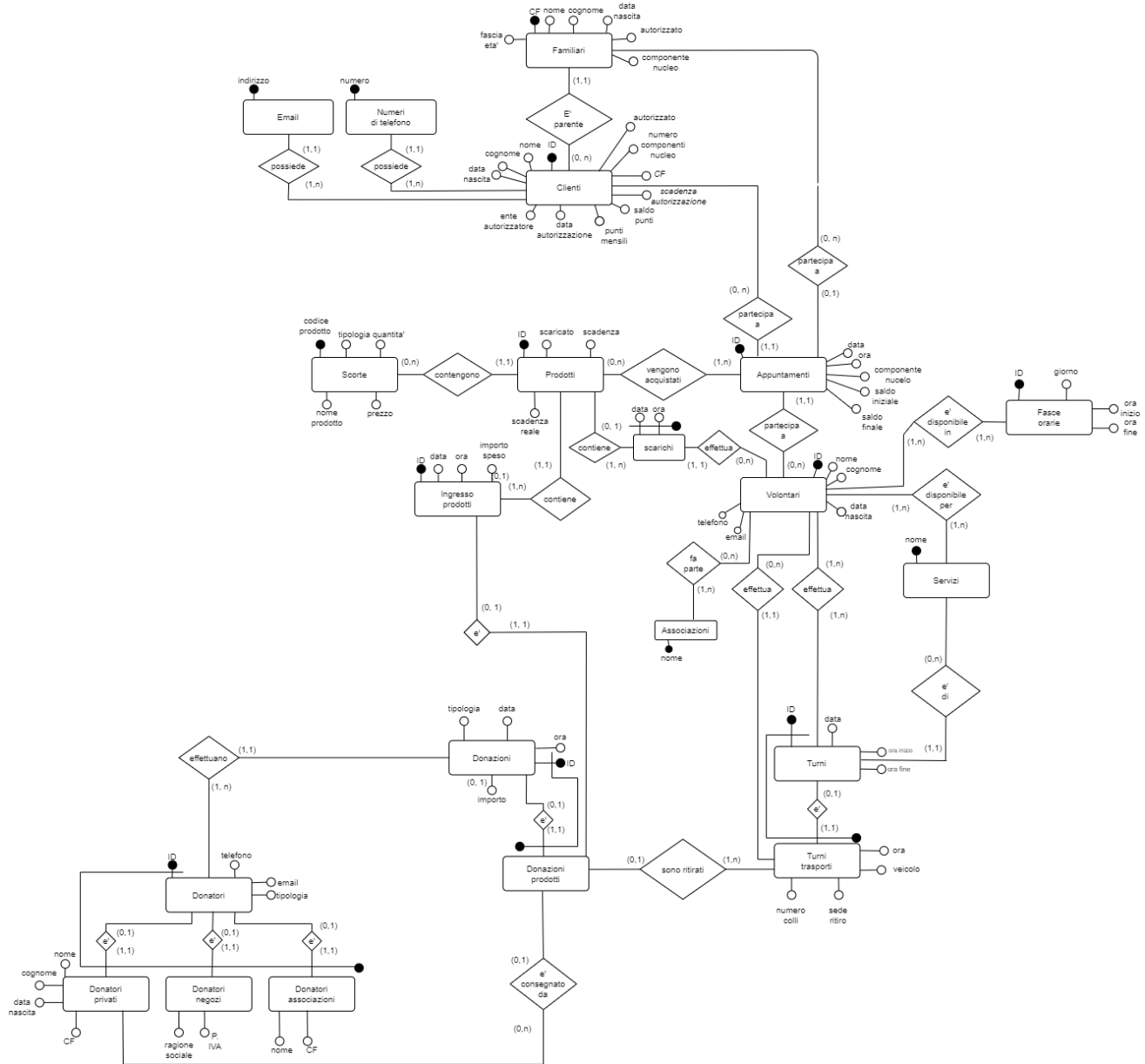


Figure 1: Diagramma ER ristrutturato

eliminazione delle gerarchie

- Servizi

Abbiamo pensato fosse sensato immagazzinare i dati del figlio nel padre (come attributi opzionali), siccome l'ID del padre e' la chiave anche del figlio, nel caso si fosse deciso di tenere 2 tabelle separate, si sarebbe dovuto fare un join per accedere ai dati del veicolo.

- Turni

Siccome sia padre che figlio sono associate ad altre tabelle, e' stato necessario "mantenere" le differenze e quindi abbiamo eliminato la gerarchia in favore di 2 tabelle associate

- Donazioni

In questo caso abbiamo optato per una soluzione ibrida, ovvero eliminare il figlio nel caso di **Donazioni** -> **denaro** (siccome non e' associata a niente) e mantenere 2 tabelle associate per **Donazioni** -> **prodotti** (visto che e' associata a diverse tabelle).

- Donatori

In questo caso per tutti e 3 i figli abbiamo associato una tabella aggiuntiva perche', anche se alcune non sono associate con niente, ci sembra piu' comodo avere memorizzati dati cosi' diversi in tabelle diverse.

modifiche effettuate prima della traduzione

Turni e servizi

Ristrutturando abbiamo notato due particolarita' dello schema ER che ci fanno storcere il naso riguardo ai **turni** e i **servizi**, ovvero:

- Non sappiamo a che servizio corrisponde un dato turno (sappiamo solo se e' un turno di trasporti ma non sappiamo con quale veicolo e' effettuato)
- Il veicolo viene (eventualmente) salvato in **servizi**, quindi potremmo solamente memorizzare come informazione che il servizio trasporti viene eseguito sempre con lo stesso veicolo.

Per risolvere questi problemi, e' bastato associare **turni** e **servizio** (turno 'e' di' servizio), cosi' facendo viene fuori una relazione (1, n) (dal lato di turno), avendo cosi' come chiave eseterna in turni l'ID del servizio che si sta svolgendo. Viene anche spostato il veicolo nel turno, in modo che sia collegato al singolo turno e che possano quindi essere memorizzati veicoli diversi in turni diversi.

Fasce orarie di disponibilita' dei volontari

Inizialmente abbiamo pensato di semplicemente scrivere le fasce orarie per ogni volontario come stringa, pero' questa soluzione implica che il volontario sia disponibile in una sola fascia oraria ('giovedi dalle 15 alle 17') oppure scrivere una stringa piu' lunga scrivendo le varie disponibilita' separate da virgole, con cui pero' sarebbe stato difficile lavorare. Ci sembra ragionevole ristrutturare quindi l'attributo "disponibilita'" in una tabella aggiuntiva "fasce_orarie" con un ID come chiave, cosi' da poter associare piu' fasce orarie ai singoli volontari e da poter controllare piu' facilmente, per esempio, che un volontario non abbia un turno assegnato in un orario in cui non e' disponibile.

familiari e appuntamenti

Per come lo abbiamo ora, il nostro database ci permette di memorizzare solamente gli appuntamenti a cui ha partecipato un cliente, ma non i suoi familiari. Siccome memorizziamo se un familiare e' autorizzato o meno ad accedere al market, e' ragionevole possa quindi partecipare agli appuntamenti. Quindi aggiungiamo un'ulteriore associazione **Familiari** 'partecipa a' **Appuntamenti**, con cardinalita' (1, n) dal lato di appuntamenti, risultandone una chiave esterna in appuntamenti. Notiamo che questa chiave esterna e' pero' opzionale, mentre invece la chiave esterna "cliente" in appuntamenti la manterremo (per poter memorizzare quale autorizzazione il familiare ha usato).

schema logico

Familiari(CF, nome, cognome, data_nascita, autorizzato, componente_nucleo, fascia_eta, cliente^{clienti})

Clienti(ID, nome, cognome, data_nascita, ente_autorizzatore, data_autorizzazione, scadenza_autorizzazione, punti_mensili, saldo_punti, CF, n_componenti_nucleo, autorizzato)

Telefoni(numero, cliente^{clienti})

Email(indirizzo, cliente^{clienti})

Appuntamenti(ID, data, ora, componente_nucleo, saldo_iniziale, saldo_finale, cliente^{clienti}, volontario^{volontari}, familiare^{familiari}_O)

Prodotti(ID, scadenza_o, scadenza_reale_o, codice_prodotto^{scorte}, ID_ingresso^{ingresso_prodotti}, data_scarico^{scarichi}_o, ora_scarico^{scarichi}_o scaricato)

Scorte(codice_prodotto, tipologia, marca, prezzo, quantita')

Scarichi(data, ora, volontario^{volontari}_o)

Ingresso_prodotti(ID, data, ora)

Volontari(ID, nome, cognome, data_nascita, telefono, email)

Fasce_orarie(ID, giorno, ora_inizio, ora_fine)

Associazioni(nome)

Servizi(nome)

Turni(ID, data, ora_inizio, ora_fine, servizio^{servizi})

Turno_trasporti(ID^{turni}, volontario^{volontario}, ora, n_colli, veicolo, sede_ritiro)

Donazioni(ID, tipologia, data, ora, importo_o donatore^{donatori})

Donazioni_prodotti(donazione^{donazioni}, consegnatario_privato^{donatori_privati}_o, ID_turno_consegna^{turni_trasporti}_o, ID_ingresso^{ingresso_prodotti})

Donatori(ID, telefono, email, tipologia)

Donatori_privati(ID^{donatori}, nome, cognome, data_nascita, CF)

Donatori_negozzi(ID^{donatori}, ragione_sociale, p_iva)

Donatori_associazioni(ID^{donatori}, nome, CF)

associazioni (n,n)

appuntamenti_prodotti(prodotto^{prodotti}, appuntamento^{appuntamenti})

volontari_associazioni(volontario^{volontari}, associazione^{associazioni})

volontari_turni(volontario^{volontari}, turno^{turni})

volontari_servizi(volontario^{volontari}, servizio^{servizi})

volontari_fasce_orarie(volontario^{volontari}, fascia_oraria^{fasce_orarie})

Normalizzazione

Per verificare la qualita' dello schema ER ristrutturato e' bene controllare che rispetti la **forma normale di Boyce Codd** e, nel caso non la rispettasse e non fosse possibile decomporre lo schema in modo da fargliela rispettare, la **terza forma normale** (che invece e' sempre possibile). Cominciamo elencando le dipendenze funzionali

- Familiari
 - CF → nome, cognome, data_nascita
- Clienti
 - ID → nome, cognome, data_nascita, ente_autorizzatore, data_autorizzazione, punti_mensili, saldo_punti, CF, autorizzato, n_componenti_nucleo
 - CF → nome, cognome, data_nascita
- Appuntamenti
 - ID → data, ora, componente_nucleo, saldo_iniziale, saldo_finale
 - data, ora → ID, componente_nucleo, saldo_iniziale, saldo_finale
- Prodotti
 - ID → nome, prezzo, scadenza, scadenza_reale, scaricato

- Scorte
 - $\text{codice_prodotto} \rightarrow \text{tipologia, quantita'}$
 - $\text{tipologia, marca} \rightarrow \text{prezzo}$
- Scarichi
 - $\text{data, ora} \rightarrow \text{volontario}^{\text{volontario}}$
- Ingresso_prodotti
 - $ID \rightarrow \text{data, ora}$
- Acquisto
 - $ID_ingresso^{\text{ingresso_prodotti}} \rightarrow \text{importo_speso}$
- Volontari
 - $ID \rightarrow \text{nome, cognome, data_nascita, telefono, email, disponibilita'}$
 - $\text{telefono} \rightarrow ID$
 - $\text{email} \rightarrow ID$
- Fasce orarie
 - $ID \rightarrow \text{giorno, ora_inizio, ora_fine}$
- Turni
 - $ID \rightarrow \text{data, ora_inizio, ora_fine}$
- Turno_trasporti
 - $ID^{\text{turni}} \rightarrow \text{volontario}^{\text{volontari}}, \text{ora, n_colli, sede_ritiro}$
- Donazioni
 - $ID \rightarrow \dots$
 - $\text{data, ora} \rightarrow ID$
- Donazioni_denaro
 - $\text{donazione}^{\text{donazioni}} \rightarrow \text{importo_speso}$
- Donazioni_prodotti
 - $\text{donazione}^{\text{donazioni}} \rightarrow \dots$
 - $ID_ingresso^{\text{ingresso_prodotti}} \rightarrow \text{donazione}^{\text{donazioni}}$
- Donatori
 - $ID \rightarrow \dots$
 - $\text{telefono} \rightarrow ID, \text{email}$
 - $\text{email} \rightarrow ID, \text{telefono}$
- Donatori_privati
 - $ID^{\text{donatori}} \rightarrow \dots$
 - $CF \rightarrow \text{nome, cognome, data_nascita}$
- Donatori_negozii
 - $ID^{\text{donatori}} \rightarrow \dots$
 - $p_iva \rightarrow ID^{\text{donatori}}$
- Donatori_associazioni
 - $ID^{\text{donatori}} \rightarrow \dots$
 - $CF \rightarrow ID^{\text{donatori}}$

Si puo' notare che tutte le dipendenze "sinistre" contengono una chiave, di conseguenza lo schema e' normalizzato rispetto a Boyce Codd

NOTA: " $ID \rightarrow \dots$ " indica che l'ID implica tutti gli altri attributi della relazione (essendo chiave)