

Progetto concettuale

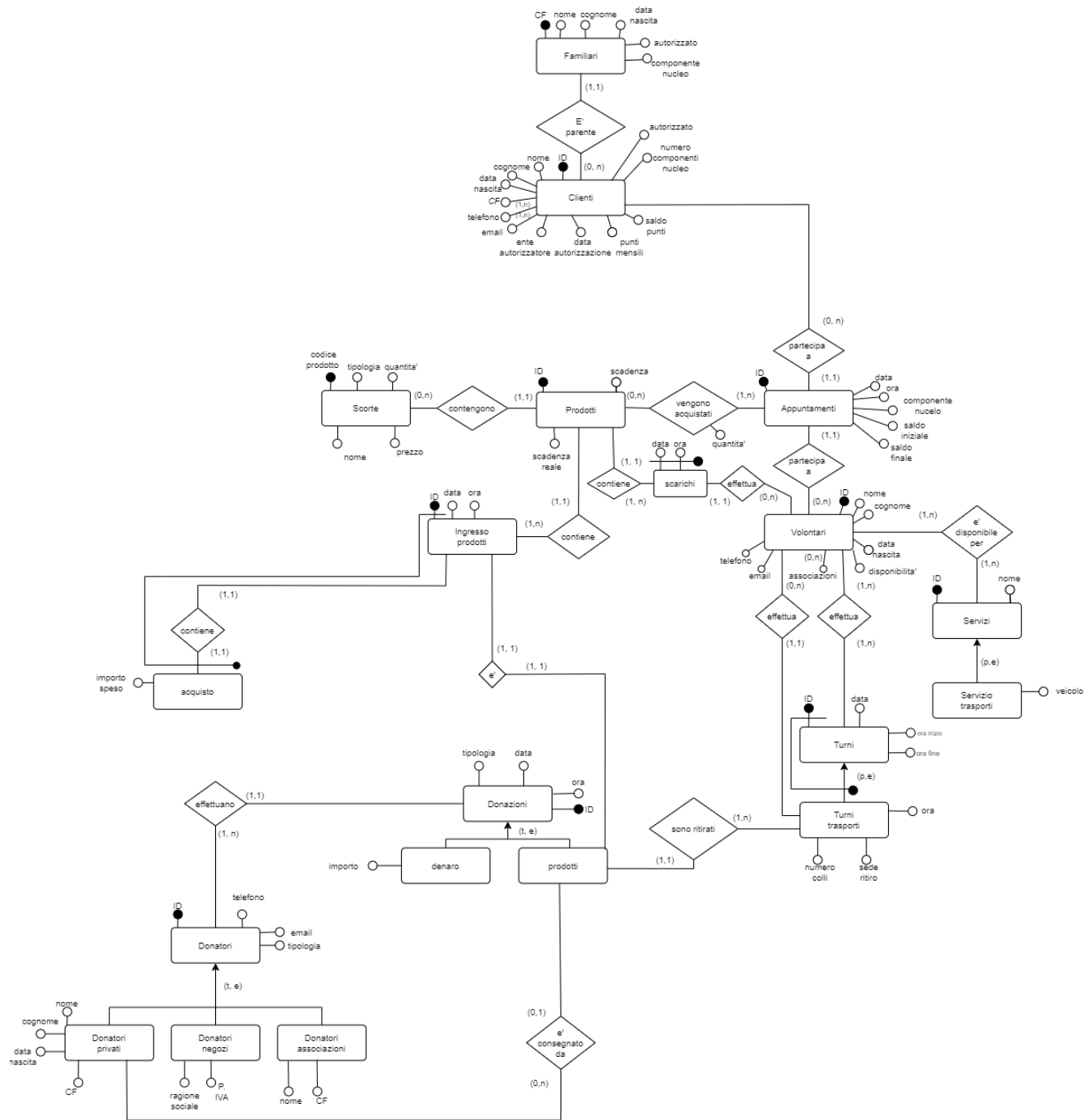


Figure 1: Diagramma ER

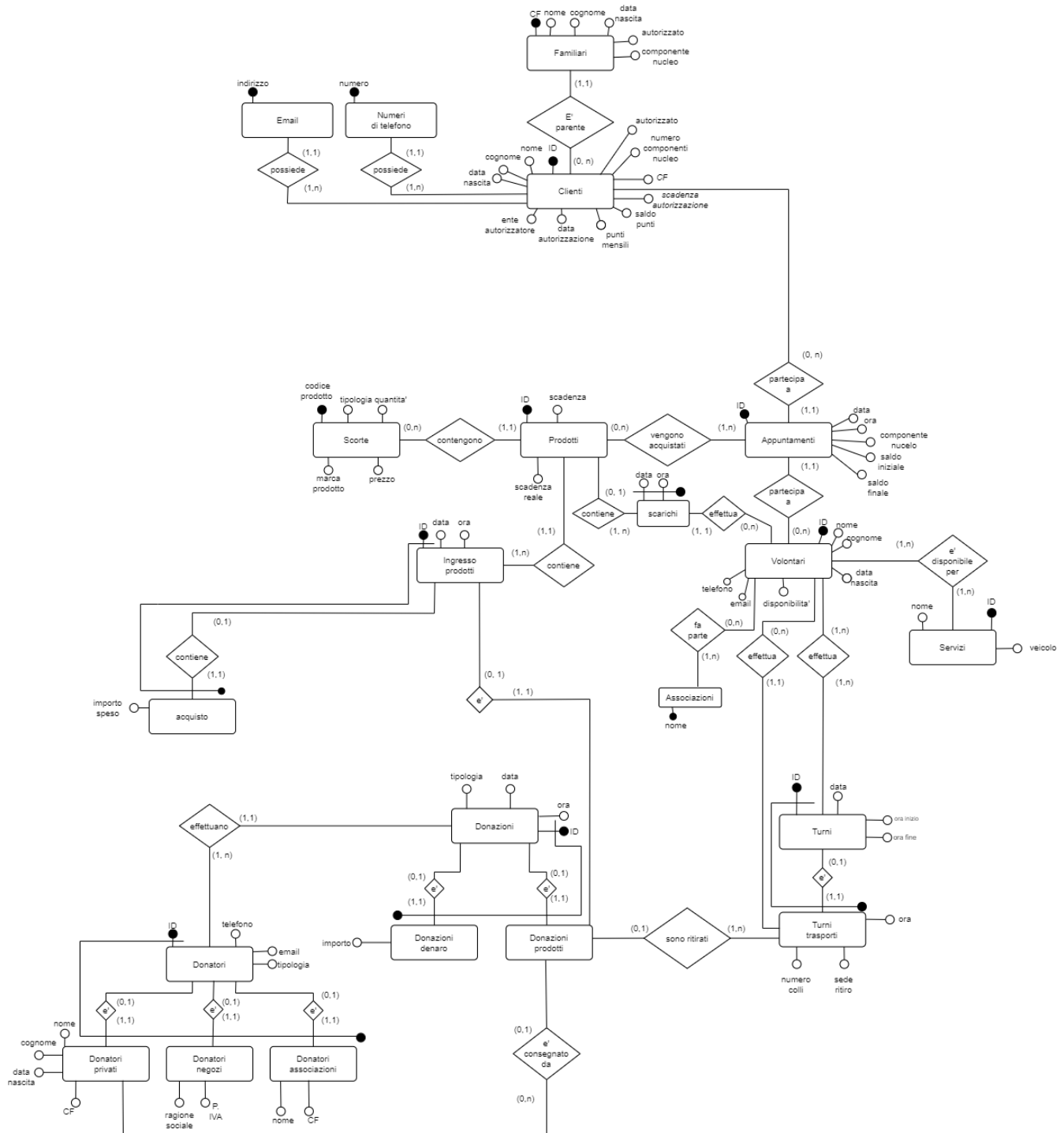


Figure 2: Diagramma ER ristrutturato

Dizionario entita'

- Telefoni
 - **numero:** **string**
 - cliente^{clienti}: **int**
 - * identificativo del cliente a cui e' collegato il numero di telefono
- Email
 - **indirizzo:** **string**
 - cliente^{clienti}: **int**
 - * identificativo del cliente a cui e' collegato l'indirizzo email
- Clienti
 - **ID:** **int**
 - * Numero identificativo (unico per ogni cliente)
 - **nome:** **string**
 - **cognome:** **string**
 - **data di nascita:** **date**
 - **codice fiscale:** **string**
 - **ente_autorizzatore:** **string**
 - * L'ente che ha concesso l'autorizzazione al cliente
 - **data_autorizzazione:** **date**
 - * data di conseguimento dell'autorizzazione
 - **scadenza_autorizzazione:** **date**
 - * di default dopo 6 mesi dalla data di autorizzazione
 - **punti_mensili:** **int**
 - * saldo mensile che ogni cliente puo' spendere
 - **saldo_punti:** **int**
 - * saldo punti attuale
 - **n_componenti_nucleo:** **int**
 - * il numero dei componenti del nucleo familiare
 - **autorizzato:** **bool**
 - * se il cliente e' autorizzato a spendere i punti oppure no
- Familiari
 - **CF:** **string**
 - * codice fiscale
 - **nome:** **string**
 - **cognome:** **string**
 - **data_nascita:** **date**
 - **componente_nucleo:** **string**
 - * quale componente del nucleo familiare e' (padre, madre, figlio...)
 - **autorizzato:** **bool**
 - * se e' autorizzato a spendere i punti oppure no
- Volontari
 - **ID:** **int**
 - * Numero identificativo del volontario (unico per ogni volontario)
 - **nome:** **string**
 - **cognome:** **string**
 - **data di nascita:** **date**
 - **telefono:** **string**
 - * unico per ogni volontario
 - **email:** **string**
 - * unico per ogni volontario
 - **disponibilita':** **string**
 - * fascia oraria e giorni in cui e' disponibile per i servizi (es. il giovedi' dalle 3 alle 5)
- Associazioni
 - **nome:** **string**
- Prodotti
 - **ID:** **int**
 - * identificativo del singolo prodotto

- scadenza: **date**
- scadenza_reale: **date**
 - * data oltre il quale e' necessario effettuare lo scarico del prodotto
- codice_prodotto^{scorte}: **int**
 - * identificativo della categoria di prodotto
- ID_ingresso^{ingresso_prodotto}: **int**
 - * ingresso prodotti in cui il singolo prodotto e' entrato nel market
- data_scarico^{scarichi}: **date**
- ora_scarico^{scarichi}: **time**
- **Scarichi**
 - data: **date**
 - ora: **time**
 - volontario^{volontari}: **int**
- **Scorte**
 - codice_prodotto: **int**
 - * codice identificativo per tutti i prodotti con una data tipologia e marca
 - tipologia: **string**
 - * Tipologia generica del prodotto (pasta, tonno...)
 - marca: **string**
 - * marca del prodotto (de Cecco, Rio Mare...)
 - prezzo: **float**
 - * costo in punti
 - quantita: **int**
 - * Quantita' disponibile di un dato prodotto in magazzino
- **Ingresso_prodotto**
 - ID: **int**
 - data: **date**
 - ora: **time**
- **Acquisto**
 - ID_ingresso^{ingresso_prodotto}: **int**
 - importo_speso: **float**
- **Servizi**
 - ID: **int**
 - nome: **string**
 - * nome del servizio (es. riordino prodotti)
 - veicolo: **string**
 - * tipologia del veicolo usato nel caso di un servizio di trasporti
- **Turni**
 - ID: **int**
 - data: **date**
 - ora_inizio: **time**
 - ora_fine: **time**
- **Turni_trasporto**
 - ID^{turni}: **int**
 - data^{turni}: **date**
 - ora: **time**
 - n_colli: **int**
 - * Numero di cestelli/scatoloni da ritirare
 - sede_ritiro: **string**
- **Donazioni**
 - ID: **int**
 - data: **date**
 - ora: **time**
 - tipologia: **string**
 - * “denaro” o “prodotti”
 - donatore^{donatori}: **int**
- **Donazioni_denaro**
 - ID^{donazioni}: **int**
 - importo: **float**

- **Donazioni_prodotti**
 - **ID**^{donazioni}: int
 - **ID_ingresso**^{ingresso_prodotti}: int
 - * identificativo dell'ingresso prodotti che contiene i prodotti donati
 - **turno_trasporti**^{turni_trasporti}
 - * ID del turno durante cui si svolge il ritiro, NULL se la donazione e' da un privato
 - **consegnatario_privato**^{donatori_privato}: int
 - * ID del consegnatario privato se la donazione e' da un privato, NULL altrimenti
- **Donatori**
 - **ID**: int
 - **telefono**: string
 - **email**: string
 - **tipologia**: string
 - * "privato", "negozio" o "associazione"
- **Donatori_privati**
 - **ID**^{donatori}: int
 - **nome**: string
 - **cognome**: string
 - **data_nascita**: date
 - **CF**: string
 - * codice fiscale
- **Donatori_negozi**
 - **ID**^{donatori}: int
 - **ragione_sociale**: string
 - **p_iva**: string
- **Donatori_associazioni**
 - **ID**^{donatori}: int
 - **nome**: string
 - **CF**: string
 - * codice fiscale

associazioni (n, n)

- **appuntamenti_prodotti**
 - **prodotto**^{prodotti}: int
 - * identificativo del prodotto acquistato
 - **appuntamento**^{appuntamenti}: int
 - * appuntamento durante il quale il prodotto e' stato acquistato
 - **quantita'**
- **volontari_associazioni**
 - **volontario**^{volontari}: int
 - **associazione**^{associazioni}: string
- **volontari_turni**
 - **volontario**^{volontari}: int
 - **turno**^{turni}: int
- **volontari_servizi**
 - **volontario**^{volontari}: int
 - **servizio**^{servizi}: string

carico di lavoro

Per effettuare tutte le operazioni al meglio, e' necessario stimare un carico di lavoro (quali operazioni verranno fatte piu' spesso, il volume dei dati nel tempo...).

Essendo un social market, ci si aspetta che abbia (sfortunatamente) abbastanza clienti ma non nell'ordine delle decine di milioni, per esempio. Sapendo che la popolazione italiana e' di circa 60,262,778 e che le persone in poverta' assoluta sono circa 5,600,000 nel 2022 (dati ISTAT), in percentuale siamo sul circa 10,8%. Ora, prendendo la popolazione per esempio di Genova nello stesso anno (568,999), il 10,8% corrisponde a circa 61,451.892, approssimato diventa 61,452. In ogni caso siamo sulle decine/centiaia di migliaia (per le citta' piu' popolate) di clienti. Occorre notare che per ogni cliente in media si avra' una famiglia al seguito, quindi, supponendo che

mediamente le famiglie siano formate da 4 persone, si avra' qualche **centiaia di migliaia** $\cdot 4$, che nel caso delle citta' piu' popolose (es. Roma) esubera il milione di circa 200k. Quindi, nel caso peggiore, si avranno 1.200.000 clienti tra clienti autorizzati e i loro familiari.

Sapendo all'incirca quanti clienti si hanno, ci si potra' piu' o meno orientare per capire di quanti prodotti il market avra' bisogno, sicuramente piu' dei clienti. Quindi si suppone che, per quantita', i prodotti saranno quelli con il maggior volume tra tutti gli altri dati, seguiti dai clienti (e i loro familiari).

Si suppone che le operazioni svolte maggiormente saranno lo stoccaggio dei prodotti in inventario (quindi inserimenti di prodotti e modifiche delle quantita' nelle scorte), quindi bisogna cercare di non sprecare memoria (per esempio con colonne a **null**) e bisogna ottimizzare le operazioni in particolare su questi dati. Ovviamente anche le altre operazioni (es. creazione turni) verranno fatte regolarmente, pero' non avranno mai milioni di righe come per i clienti o i prodotti in inventario.

Schema logico

Familiari(CF, nome, cognome, data_nascita, autorizzato, componente_nucleo, 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})

Prodotti(ID, scadenza_o, scadenza_reale_o, codice_prodotto^{scorte}, ID_ingresso^{ingresso_prodotti}, data_scarico_o^{scarichi}, ora_scarico_o^{scarichi})

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

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

Ingresso_prodotti(ID, data, ora)

Acquisto(ID_ingresso^{ingresso_prodotti}, importo_speso)

Volontari(ID, nome, cognome, data_nascita, *telefono*, *email*, disponibilita')

Associazioni(nome)

Servizi(ID, nome, veicolo_o)

Turni(ID, data, ora_inizio, ora_fine)

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

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

Donazioni_denaro(donazione^{donazioni}, importo)

Donazioni_prodotti(donazione^{donazioni}, consegnatario_privato_o^{donatori_privati}, ID_turno_consegna_o^{turni_trasporti}, ID_ingresso^{ingresso_prodotti})

Donatori(ID, *telefono*, *email*, tipologia)

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

Donatori_negozi(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})

Normalizzazione

Per verificare la qualità dello schema ER ristrutturato è 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 è sempre possibile). Cominciamo elencando le dipendenze funzionali

- Familiari(CF, nome, cognome, data_nascita, cliente^{clienti})
 - $CF \rightarrow \text{nome, cognome, data_nascita}$
- Clienti
 - $ID \rightarrow \text{nome, cognome, data_nascita, ente_autorizzatore, data_autorizzazione, punti_mensili, saldo_punti, CF, autorizzato, n_componenti_nucleo}$
 - $CF \rightarrow \text{nome, cognome, data_nascita}$
- Appuntamenti
 - $ID \rightarrow \text{data, ora, componente_nucleo, saldo_iniziale, saldo_finale}$
 - $\text{data, ora} \rightarrow ID, \text{componente_nucleo, saldo_iniziale, saldo_finale}$
- Prodotti
 - $ID \rightarrow \text{nome, prezzo, scadenza, scadenza_reale}$
- 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
 - $ID \rightarrow \text{nome, cognome, data_nascita, telefono, email, disponibilita'}$
 - $\text{telefono} \rightarrow ID$
 - $\text{email} \rightarrow ID$
- Servizi
 - $ID \rightarrow \text{nome, veicolo}$
- 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 può notare che tutte le dipendenze “sinsitre” contengono una chiave, di conseguenza lo schema è normalizzato rispetto a Boyce Codd ## Query

Tutti i prodotti acquistati durante l'ultimo appuntamento del cliente con ID = 1

```
SELECT *  
FROM Prodotti  
JOIN Appuntamenti_prodotti  
ON Appuntamenti_prodotti.prodotto = prodotti.ID  
JOIN Appuntamenti  
ON Appuntamenti_prodotti.appuntamento = Appuntamenti.ID  
JOIN Clienti  
ON Appuntamenti.cliente = Clienti.ID  
WHERE Clienti.ID = 1
```