Relazione Laboratorio Basi Di Dati

Voltan Gabriele, Pagnutti Emanuele, Castenetto Davide

March 2022

1 Introduzione

Lo scopo dell'attività di laboratorio è quello di analizzare, progettare e implementare una base di dati per il dominio applicativo assegnato.

Il progetto è così composto:

- ***SQL***:
  - **codiceSQL.sql** contenente la definizione delle tabelle, dei trigger e degli indici
  - **operazioni.sql** contenente le query delle operazioni più frequenti
- ***Python***:
  - **main.py** contenente il codice python per la generazione del file .sql per il popolamento della base di dati
  - **PopolamentoInizialeTabelle.sql** contenente le operazione di insert e update
  - **Prodotti.csv** contenente un elenco di nomi dei prodotti
  - **Reparti.csv** contenente un elenco di nomi dei reparti
- ***R***
  - **main.r** contente il codice per la generazione dei grafici

2 Analisi dei requisiti

Il testo del progetto assegnato è il seguente:

Si vuole progettare una base di dati di supporto alla gestione delle informazioni di interesse per un mercato di merci all'ingrosso, contenente le informazioni sotto specificate.

- Un aspetto fondamentale è la gestione delle scorte. Il sistema deve memorizzare informazioni relative ai fornitori e ai prodotti da questi forniti. Di ogni fornitore, vogliamo memorizzare il nome, l'indirizzo, i recapiti telefonici e i prodotti forniti. Si assuma che uno stesso prodotto possa essere fornito da più fornitori e che un fornitore possa fornire più prodotti. Di ogni prodotto, vogliamo memorizzare il nome, il codice prodotto e il prezzo di vendita. Si assuma che un dato prodotto venga identificato univocamente dal codice, indipendentemente dai fornitori che lo forniscono. Vogliamo inoltre tener traccia del prezzo di fornitura dei prodotti da parte dei diversi fornitori.

- Per quanto riguarda l'organizzazione del mercato, si assuma che esso sia strutturato in reparti. Ogni reparto è identificato da un nome e da un numero. Ogni reparto è gestito da un capo reparto e dispone di un insieme di dipendenti. Si assuma che ogni dipendente afferisca ad un unico reparto. Di ogni dipendente, vogliamo memorizzare il nome, il codice fiscale, l'indirizzo, lo stipendio e eventuali altre informazioni di interesse. Ogni reparto è responsabile dell'acquisto di alcuni prodotti. Si assuma che ogni ordine d'acquisto sia caratterizzato da un numero d'ordine (tale numero identifica univocamente l'ordine nell'insieme degli ordini fatti dal reparto), dal codice prodotto, dall'ammontare di prodotto richiesto, dalla data dell'ordine e dal fornitore prescelto.

- Vogliamo infine, mantenere informazioni sui clienti del mercato. Di ogni cliente, vogliamo memorizzare il nome, l'indirizzo e il saldo delle spese. I clienti ordinano i prodotti che vengono consegnati loro dal mercato. Ogni ordine è contraddistinto da un numero d'ordine (tale numero identifica univocamente l'ordine nell'insieme degli ordini fatti dal cliente), da una data e dall'insieme dei prodotti ordinati, con l'indicazione, per ciascuno di essi, dell'ammontare richiesto.

Il **dominio applicativo** della base di dati è la *gestione delle informazioni per un mercato di merci all'ingrosso.*

2.1 Glossario dei termini

Termine	Descrizione	Sinonimi	Collegamenti
Fornitore	Fornisce i prodotti richiesti dai vari reparti		Prodotto, Ordine reparto
Prodotto	Elemento richiesto e ordinato dai clienti o dai reparti		Fornitore, Reparto, Ordine Reparto, Ordine Cliente
Reparto	Parte dell'organizzazione del mercato		Prodotto, Ordine Reparto, Dipendenti
Dipendenti	Può essere o un impiegato semplice o capo reparto		Reparto
Cliente	Persona che effettua un ordine al mercato		Ordine cliente
Ordine Cliente	Ciò che richiede il cliente		Cliente, Prodotto
Ordine Reparto	Ciò che richiede il reparto		Fornitore, Reparto, Prodotto

2.2 Riscrittura e strutturazione dei requisiti

Frasi di carattere generale

Un aspetto fondamentale è la gestione delle scorte. Il sistema deve memorizzare informazioni relative ai fornitori e ai prodotti da questi forniti.
Per quanto riguarda l'organizzazione del mercato, si assuma che esso sia strutturato in reparti.

Frasi relative ai fornitori

Di ogni fornitore, vogliamo memorizzare il nome, l'indirizzo, i recapiti telefonici e i prodotti forniti. Si assuma che uno stesso prodotto possa essere fornito da più fornitori e che un fornitore possa fornire più prodotti.

Frasi relative ai prodotti

Di ogni prodotto, vogliamo memorizzare il nome, il codice prodotto e il prezzo di vendita. Si assuma che un dato prodotto venga identificato univocamente dal codice, indipendentemente dai fornitori che lo forniscono. Vogliamo inoltre tener traccia del prezzo di fornitura dei prodotti da parte dei diversi fornitori.

Frasi relative ai reparti

Ogni reparto è identificato da un nome e da un numero. Ogni reparto è gestito da un capo reparto e dispone di un insieme di dipendenti.
Ogni reparto è responsabile dell'acquisto di alcuni prodotti.

Frasi relative ai dipendenti

Si assuma che ogni dipendente afferisca ad un unico reparto. Di ogni dipendente, vogliamo memorizzare il nome, il codice fiscale, l'indirizzo, lo stipendio e eventuali altre informazioni di interesse.

Frasi relative agli ordini dei reparti

Ogni reparto è responsabile dell'acquisto di alcuni prodotti. Si assuma che ogni ordine d'acquisto sia caratterizzato da un numero d'ordine (tale numero identifica univocamente l'ordine nell'insieme degli ordini fatti dal reparto), dal codice prodotto, dall'ammontare di prodotto richiesto, dalla data dell'ordine e dal fornitore prescelto.

Frasi relative ai clienti
Vogliamo infine, mantenere informazioni sui clienti del mercato. Di ogni cliente, vogliamo memorizzare il nome, l'indirizzo e il saldo delle spese. I clienti ordinano i prodotti che vengono consegnati loro dal mercato.

Frasi relative agli ordini dei clienti
Ogni ordine è contraddistinto da un numero d'ordine (tale numero identifica univocamente l'ordine nell'insieme degli ordini fatti dal cliente), da una data e dall'insieme dei prodotti ordinati, con l'indicazione, per ciascuno di essi, dell'ammontare richiesto.

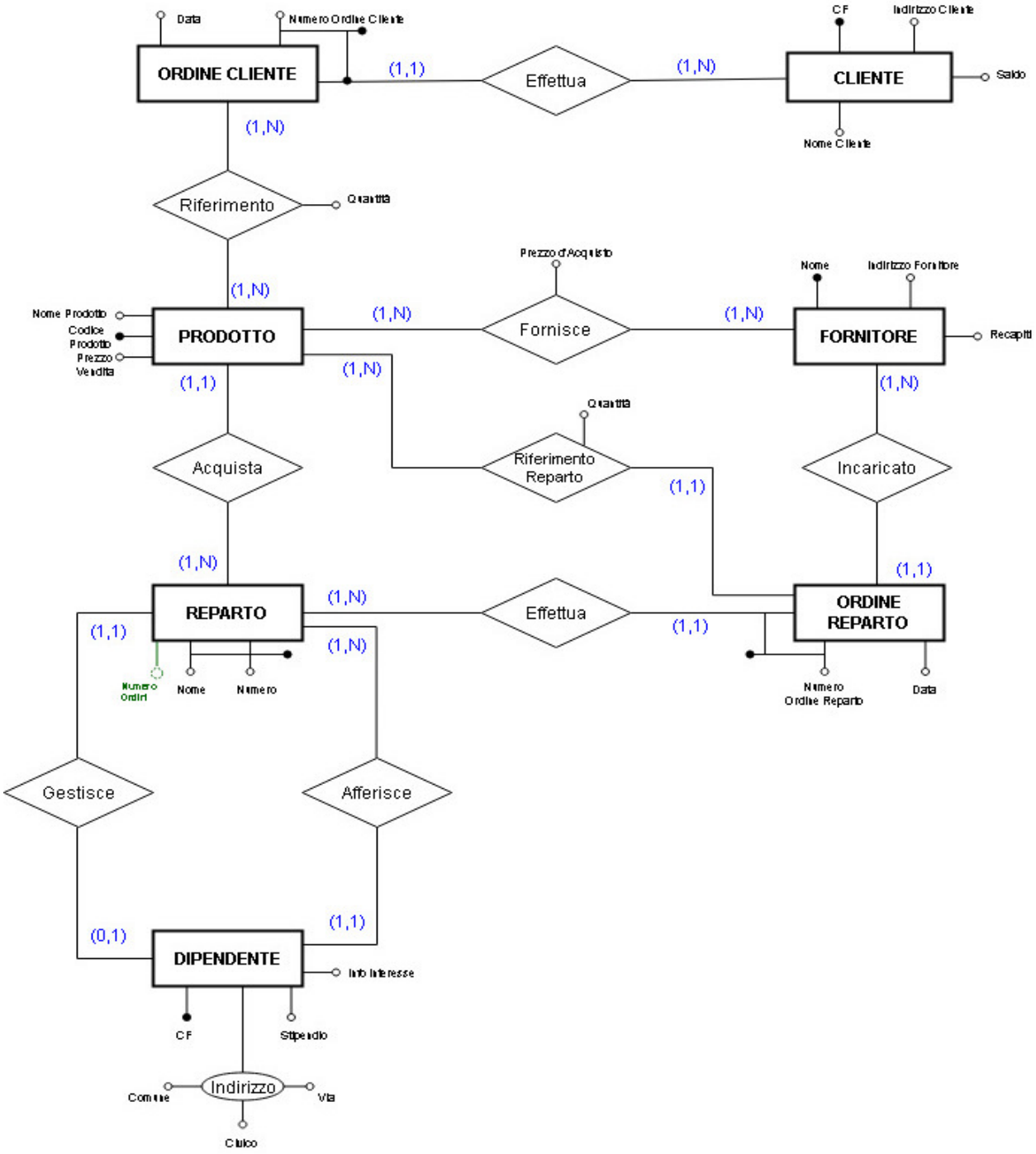
2.3 Operazioni effettuate sui dati

E' riportato di seguito l'elenco delle operazioni, con la loro rispettiva frequenza.

1. Inserimento nuovi clienti con i relativi dati personali (10 al giorno)
2. Inserimento ordini clienti con specifica prodotti (50 al giorno)
3. Inserimento ordini reparti con specifica del singolo prodotto (100 al giorno)
4. Per ogni reparto, mostrare il numero di ordini effettuati (1 al giorno)
5. Per ogni reparto, mostrare l'elenco dei prodotti in vendita (1 a settimana)
6. Per ogni reparto, mostrare la somma spesa in stipendi (1 al mese)
7. Inserimento nuova fornitura con relativo prezzo d'acquisto (50 al mese)
8. Mostrare i clienti con un saldo superiore a 50 euro (1 a settimana)

3 Progettazione concettuale

3.1 Schema Entità-Relazioni



3.1.1 Vincoli aggiuntivi

Nello schema E-R è presente un attributo derivato.
La **regola di derivazione** è la seguente:

- L'attributo ***numero ordini*** dell'entità *Reparto* è derivabile contando le istanze dell'entità *Ordine reparto* riferite al reparto in questione.

3.1.2 Decisioni progettuali e considerazioni

Nel corso della progettazione concettuale, abbiamo preso delle decisioni in merito ad alcuni aspetti non specificati nei requisiti della consegna.

- La prima decisione riguarda il vincolo di cardinalità della relazione *gestisce*: un **Dipendente** può gestire al massimo un **Reparto**.
- La seconda decisione riguarda l'aggiunta dell'attributo *Codice Fiscale* nell'entità **Cliente**: questa scelta risolve, in maniera ottima, eventuali casi di omonimia.

Un altro aspetto da trattare è la presenza dei cicli nello schema E-R, in quanto potrebbero portare a situazioni poco sensate. I cicli presenti, sono i seguenti e vengono così gestiti:

- Dipendente - Afferisce - Reparto - Gestisce
  - Per evitare casi poco sensati stabiliamo che un *Dipendente* può **gestire** solo il *Reparto* alla quale **afferisce**.
- Reparto - Effettua - Ordine Reparto - Riferimento Reparto - Prodotto - Acquista
  - Per evitare casi poco sensati stabiliamo che un *Reparto* può **acquistare** un *Prodotto* solo se ha **effettuato** un *Ordine Reparto* che si **riferisce** a tale *Prodotto*.
- Fornitore - Fornisce - Prodotto - Riferimento Reparto - Ordine Reparto - Incaricato
  - Per evitare casi poco sensati stabiliamo che un *Fornitore* **fornisce** un *Prodotto* solo se esiste un *Ordine Reparto*, che si **riferisce** a tale *Prodotto*, a lui **incaricato**.
- Reparto - Effettua - Ordine Reparto - Incaricato - Fornitore - Fornisce - Prodotto - Acquista
  - Questo ciclo non è problematico, in quanto risolto mediante la gestione dei due sotto cicli che contiene.

4 Progettazione Logica

4.1 Ristrutturazione della schema Entità Relazioni

4.1.1 Analisi delle ridondanze

Per effettuare l'analisi delle ridondanze, dobbiamo definire i volumi dei dati e riportare le operazioni, con le relative frequenze, indicate al paragrafo 2.3.

Concetto	Tipo	Volume
Cliente	Entità	100
Ordine Cliente	Entità	1000
Prodotto	Entità	300
Fornitore	Entità	40
Reparto	Entità	20
Ordine Reparto	Entità	2000
Dipendente	Entità	100
Afferisce	Relazione	100
Gestisce	Relazione	20
Effettua Ordine Reparto	Relazione	2000
Effettua Ordine Cliente	Relazione	1000
Acquista	Relazione	300
Riferimento Ordine Reparto	Relazione	2000
Fornisce	Relazione	300
Riferimento Ordine Cliente	Relazione	1500
Incaricato	Relazione	2000

Operazione	Tipo	Volume
Inserimento clienti	Interattiva	10 al giorno
Inserimento ordine clienti	Interattiva	50 al giorno
Inserimento ordini reparto	Interattiva	100 al giorno
Inserimento nuova fornitura con relativo prezzo d'acquisto	Interattiva	50 al mese
Visualizza prodotti in vendita reparto	Batch	1 a settimana
Somma spese in stipendi reparto	Batch	1 al mese
Verifica numeri ordine reparto	Batch	1 al giorno
Visualizza clienti con saldo superiore a 50 euro	Batch	1 a settimana

L'unico attributo ridondante presente nello schema E-R è **Numero Ordini** dell'entità *Reparto*. Le operazioni in cui è coinvolto, sono due: per ognuna di esse, definiamo le tabelle degli accessi.

- Inserimento ordine reparto (250 al giorno)
  - Tabella degli accessi con l'attributo ridondante

Concetto	Costrutto	Accessi	Tipo
Ordine Reparto	Entità	1	Scrittura
Riferimento Reparto	Relazione	1	Scrittura
Incaricato	Relazione	1	Scrittura
Effettua	Relazione	1	Scrittura
Reparto	Entità	1	Lettura
Reparto	Entità	1	Scrittura

– Tabella degli accessi senza l'attributo ridondante

Concetto	Costrutto	Accessi	Tipo
Ordine Reparto	Entità	1	Scrittura
Riferimento Reparto	Relazione	1	Scrittura
Incaricato	Relazione	1	Scrittura
Effettua	Relazione	1	Scrittura

- Lettura del numero di ordini effettuati da un reparto
  - Tabella degli accessi con l'attributo ridondante

Concetto	Costrutto	Accessi	Tipo
Ordine Reparto	Entità	2000	Lettura

  - Tabella degli accessi senza l'attributo ridondante

Concetto	Costrutto	Accessi	Tipo
Reparto	Entità	1	Lettura

Effettuiamo ora i calcoli complessivi, dando un peso maggiore alle operazione di scrittura, in quanto risultano essere più dispendiose per la base di dati.

- Per l'operazione di Inserimento Ordine Reparto:
  - AccessiGiornalieri = (2+2+2+2+1+2) * 100 = 1100 accessi, **con ridondanza**.
  - AccessiGiornalieri = (2+2+2+2) * 100 = 800 accessi, **senza ridondanza**.
- Per l'operazione di Lettura Numero Ordini per reparto:
  - AccessiGiornalieri = 1 accesso, **con ridondanza**.
  - AccessiGiornalieri = 2000 accessi, **senza ridondanza**.

Come possiamo notare dai risultati ottenuti, è più conveniente tenere l'attributo ridondante, in quanto ci permette di effettuare molti meno accessi.

4.1.2 Eliminazione delle generalizzazioni

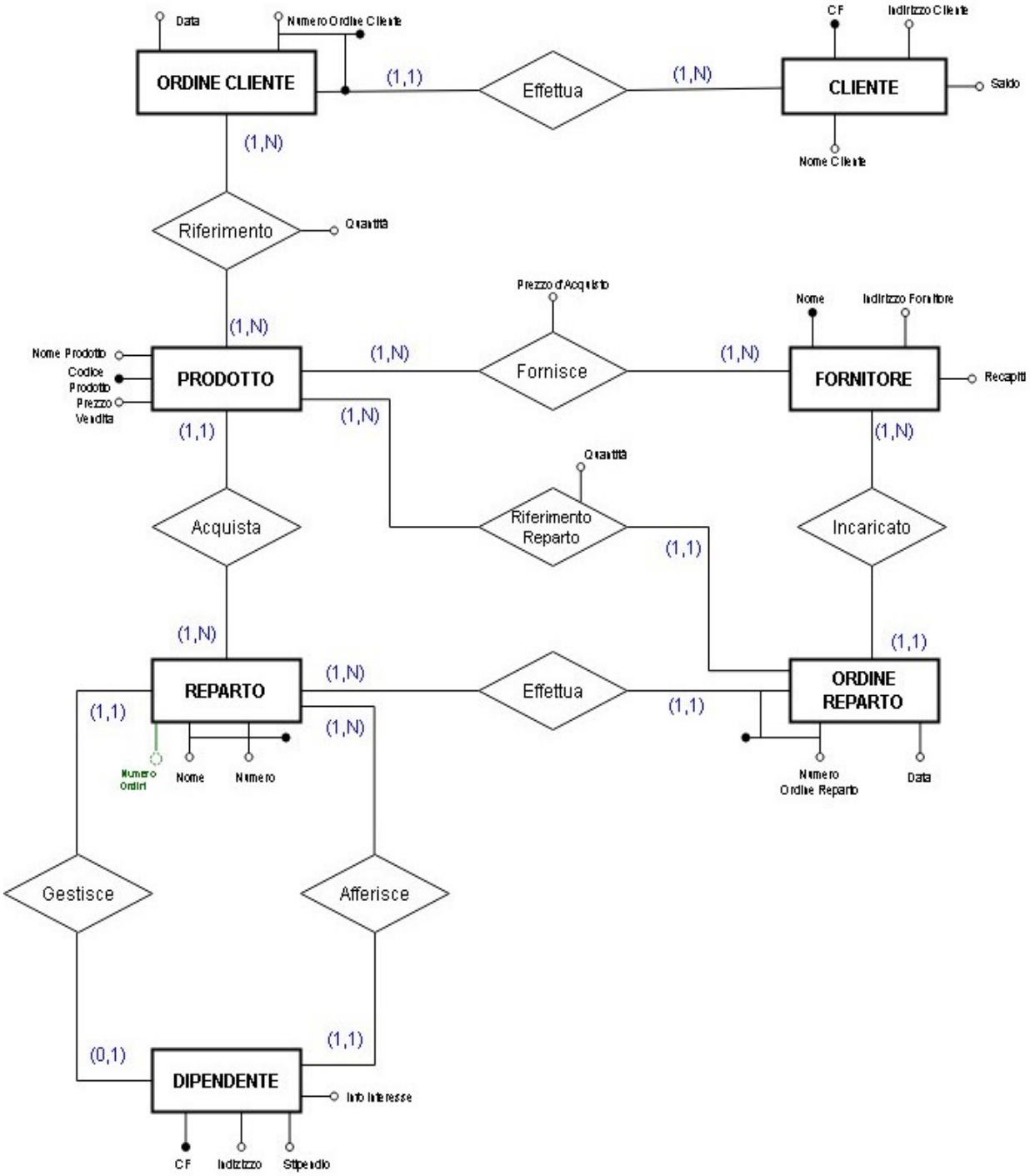
Nello schema E-R non sono presenti generalizzazioni, quindi lo schema rimane inalterato.

4.1.3 Partizionamento o merge di entità, relazioni, attributi

L'unica operazione effettuata è la rimozione dell'attributo composto **Indirizzo** dell'entità *Dipendente*. L'informazione è memorizzata in un unico attributo che conterrà l'indirizzo nella sua interezza.

4.1.4 Scelta delle chiavi primarie

Se considerassimo l'assenza di omonimie tra i clienti, un'altra chiave candidata per l'entità *Cliente* sarebbe **Nome cliente**. Essendo che però questa considerazione è poco realistica, manteniamo come chiave primaria quella indicata nello schema E-R di partenza, introdotta al paragrafo 3.1.2.



Schema E-R ristrutturato

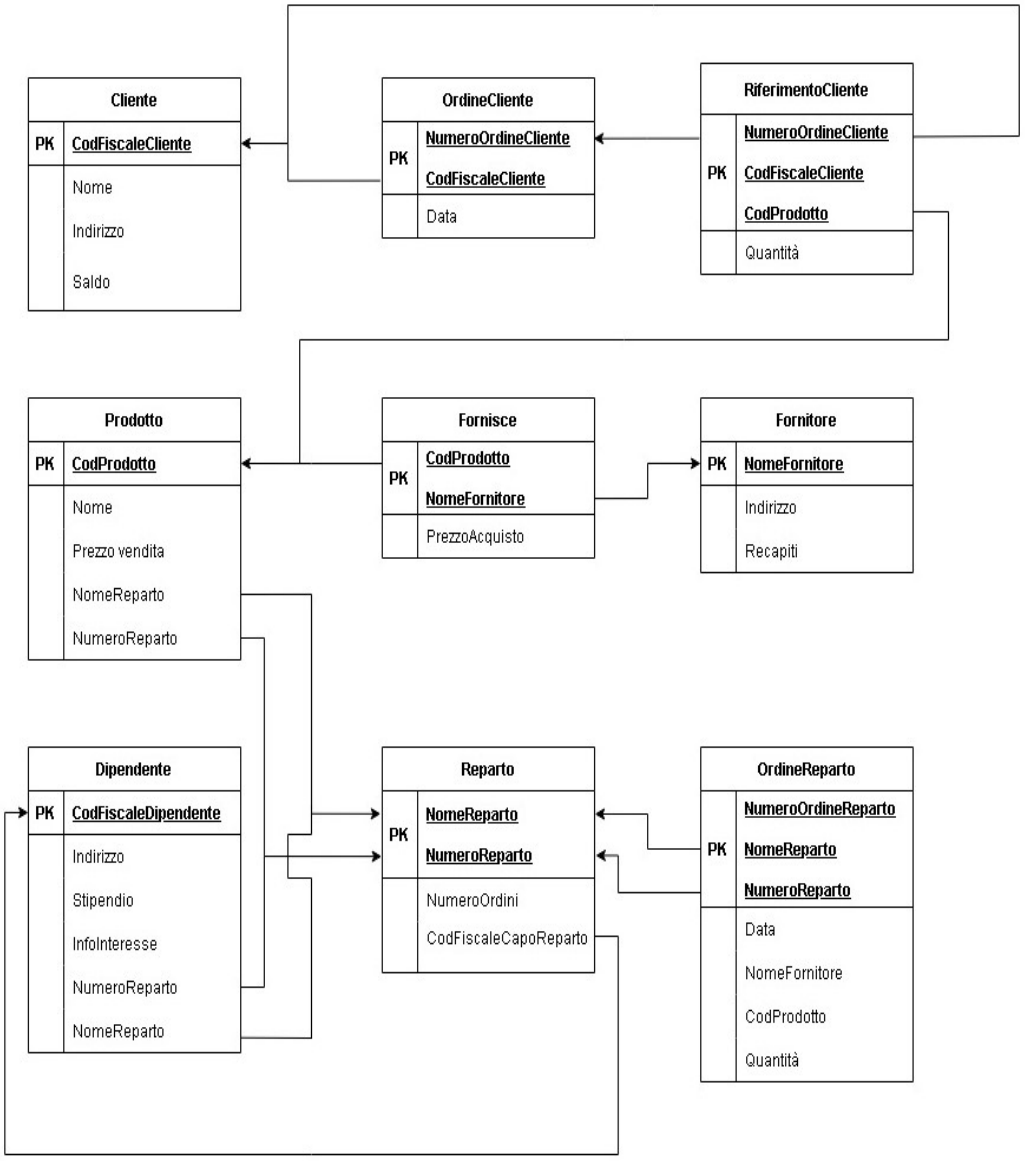
4.2 Traduzione dello schema E-R nello schema relazionale

In questa fase, traduciamo man mano i vari costrutti (entità, relazioni uno a uno, relazione uno a molti, relazioni molti a molti) dello schema E-R ristrutturato.
Lo schema relazione avrà le seguenti relazioni:

- **Cliente**(CodFiscaleCliente, Nome, Indirizzo, Saldo)
  - Vincolo NotNull: Nome, Indirizzo, Saldo
- **OrdineCliente**(*CodFiscaleCliente*, NumeroOrdineCliente, Data)
  - Vincolo NotNull: Data
- **RiferimentoCliente**(*CodFiscaleCliente, NumeroOrdineCliente, CodProdotto*, Quantità)
  - Vincolo NotNull: Quantità
- **Prodotto**(CodProdotto, Nome, PrezzoVendita, *NomeReparto, Numero Reparto*)
  - Vincolo NotNull: Nome, PrezzoVendita, NomeReparto, Numero Reparto
- **Reparto**(NomeReparto, NumeroReparto, NumeroOrdini, *CodFiscaleCapoReparto*)
  - Vincolo NotNull: NumeroOrdini, CodFiscaleCapoReparto
  - Vincolo Unique: CodFiscaleCapoReparto
- **Dipendente**(CodFiscaleDipendente, Indirizzo, Stipendio, InfoInteresse, *NomeReparto, NumeroReparto*)
  - Vincolo NotNull: Indirizzo, Stipendio, Nome Reparto, Numero Reparto
- **Fornitore**(NomeFornitore, Indirizzo, Recapiti)
  - Vincolo NotNull: Indirizzo, Recapiti
- **Fornisce**(*CodProdotto, NomeFornitore*, PrezzoAcquisto)
  - Vincolo NotNull: PrezzoAcquisto
- **OrdineReparto**(NumeroOrdineReparto, *NomeReparto, NumeroReparto*, Data, Quantità, *NomeFornitore, CodProdotto*)
  - Vincolo NotNull: Data, Quantità, Nome Fornitore, CodProdotto

4.2.1 Schema relazionale finale

Si riporta sotto lo schema relazionale finale in forma grafica.



Schema relazionale

5 Progettazione Fisica

In questa fase della progettazione ci occupiamo della realizzazione della base di dati, in maniera concreta, scrivendo il codice SQL per la definizione delle tabelle, degli indici, dei trigger e il codice Python necessario a generare il file .sql per l'inserimento dei dati.

5.1 Creazione delle tabelle

Nella parte di definizione delle tabelle, un punto interessante lo incontriamo nella definizione delle tabelle *Reparto* e *Dipendente*: essendoci una dipendenza ciclica, abbiamo dovuto prima definire la tabella del reparto, senza referenziare il codice fiscale del capo reparto. Successivamente, abbiamo definito la tabella dipendente e modificato la tabella reparto, referenziando l'attributo cfcaporeparto e dichiarandolo deferrable. In fase di inserimento quindi, prima inseriremo i reparti, poi i dipendenti e infine i capi reparti.
Il codice completo, lo si trova nel file *codiceSQL.sql*.

5.2 Inserimento dei dati casuali

Per la generazione pseudocasuale abbiamo deciso di sfruttare principalmente due risorse: il sito web *mockaroo.com* e il linguaggio *Python*.
Mockaroo.com è stato sfruttato per creare due files .csv contenenti rispettivamente i nomi dei prodotti e i nomi dei reparti.
Del linguaggio Python, abbiamo sfruttato in particolare la libreria *Faker*, in quanto questa consente la generazione pseudocasuale (attraverso un determinato seme) di diversi valori per i campi delle nostre tabelle (quali nome, cognome, indirizzo, telefono, ...). La gestione e la correttezza dei parametri derivati (ad esempio il numero ordini di un dato reparto), come anche la consistenza dei valori inseriti è svolta sempre attraverso il linguaggio Python.
Sempre utilizzando questo linguaggio abbiamo generato il file di popolamento iniziale .sql della base di dati.
Il codice python completo, lo si trova nel file main.py.

5.3 Creazione degli indici

Per migliorare le prestazioni dell'operazione *Mostrare i clienti con un saldo superiore a 50 euro* abbiamo deciso di implementare un semplice indice sul campo **saldo** dell'entità *cliente*.
L'indice lo abbiamo implementato usando i **btree**, ordinando in maniera crescente. Implementando questo indice, le prestazioni migliorano, in quando è possibile eseguire una ricerca binaria sulla tabella *Cliente*, diminuendo quindi il tempo necessario per trovare il primo valore utile per la ricerca.
Di seguito il codice sql per la creazione.

```
create index indice_saldo on cliente using btree(
  saldo ASC
);
```

5.4 Creazione dei trigger

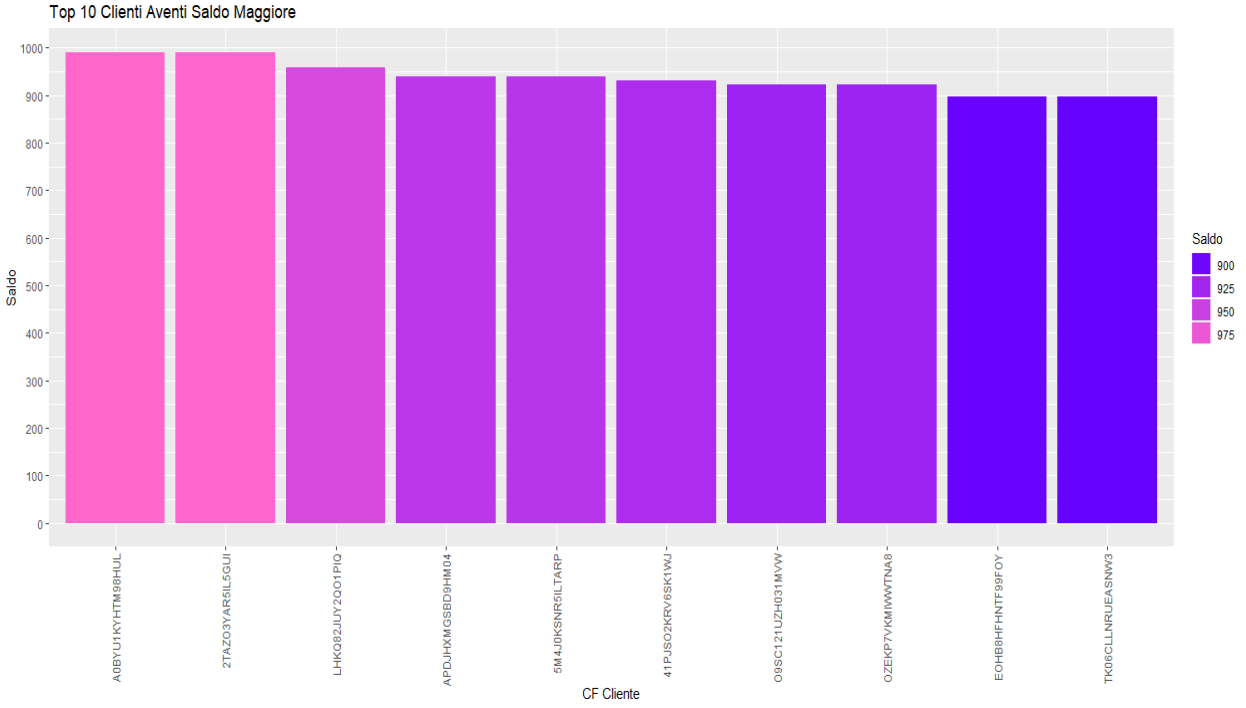
Per la gestione di alcuni vincoli, abbiamo implementato dei trigger. In particolare ne abbiamo implementati tre:

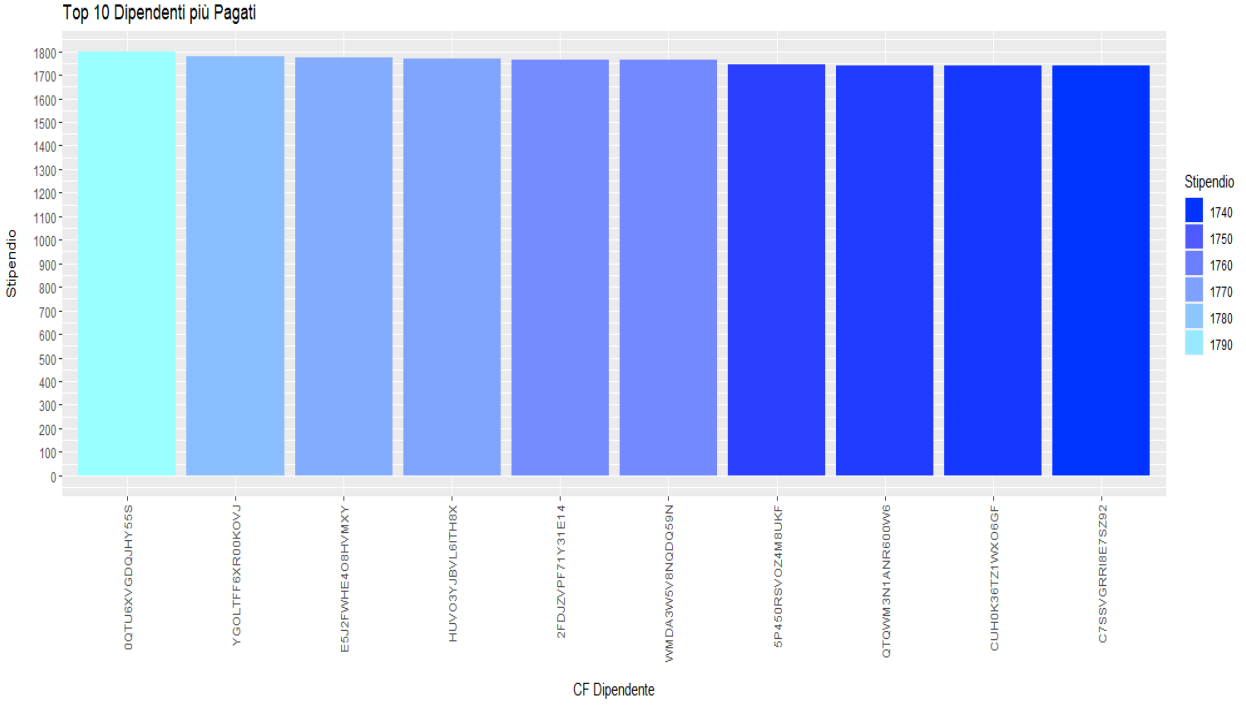
- **controlla ordini** per verificare che il prodotto ordinato da un reparto sia venduto dal reparto che lo sta cercando di ordinare.
- **check caporeparto** per verificare che il capo reparto che vogliamo inserire, afferisca al reparto di cui gli stiamo dando il comando.
- **check reparto** per verificare che il dipendente che vogliamo spostare di reparto, non sia un capo reparto.

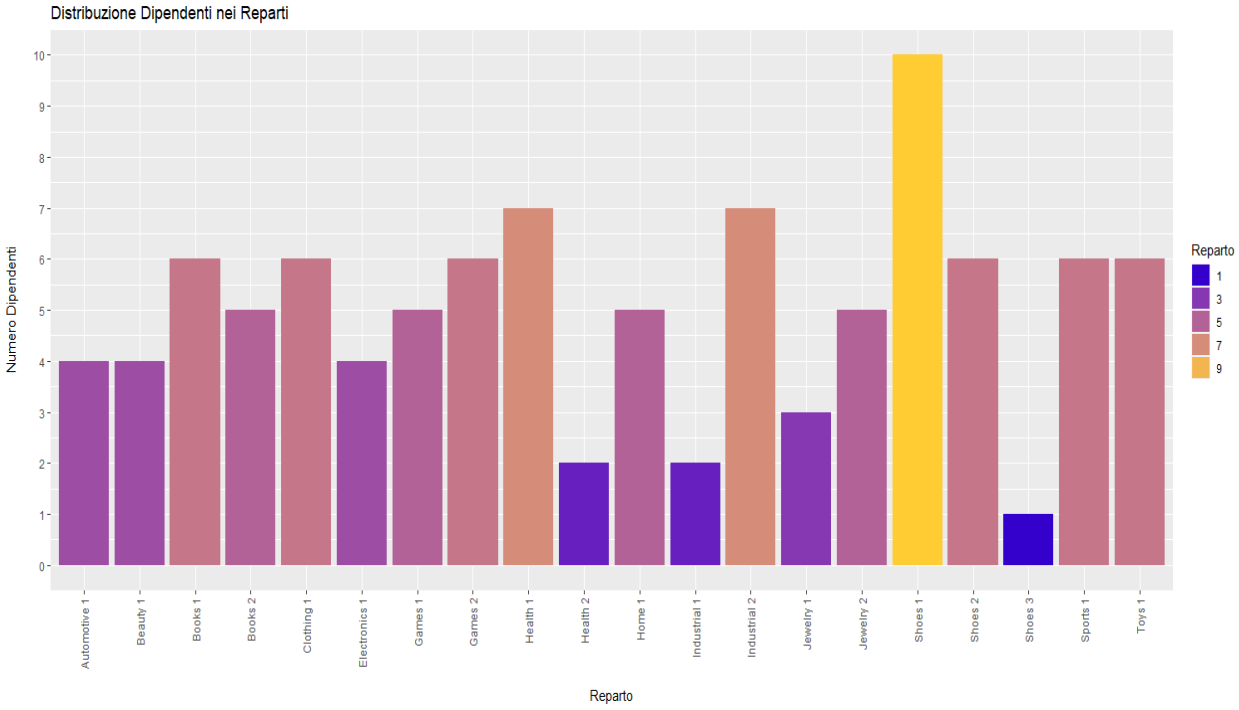
Riportiamo di seguito il codice sql del primo trigger. I codici di tutti i trigger si trovano nel file *codiceSQL.sql*.

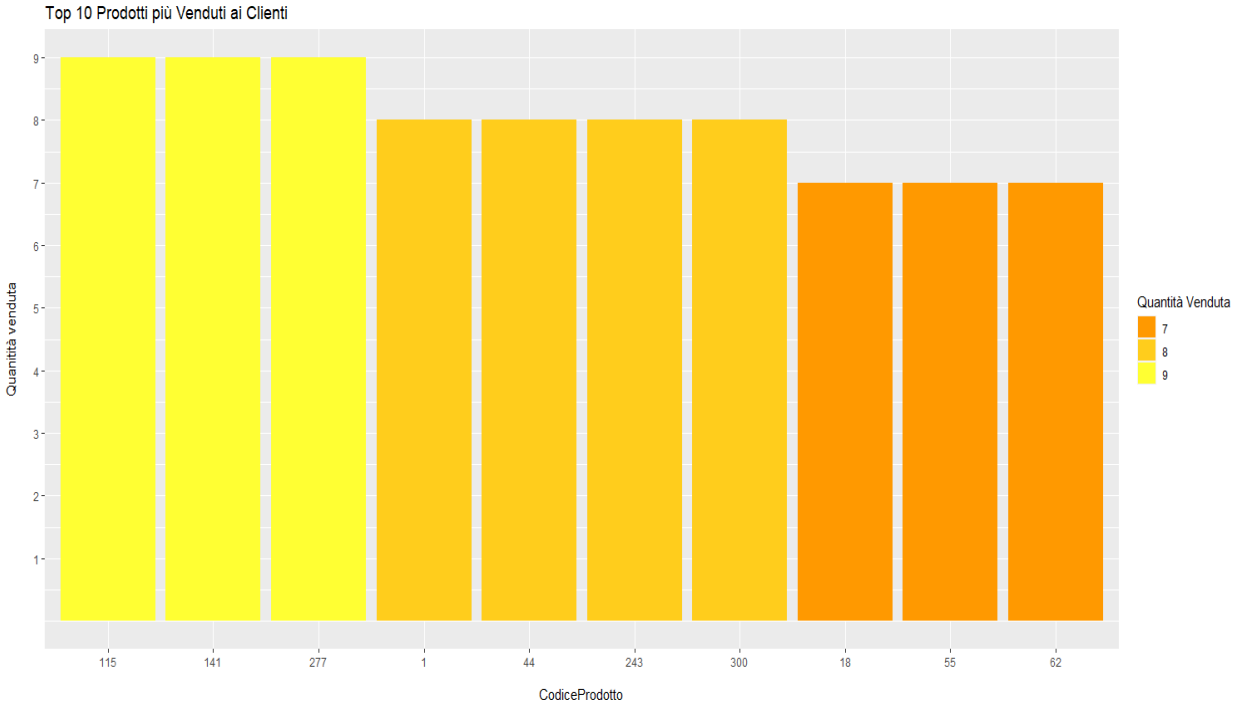
```
create or replace function check_prodotti()
  returns trigger language plpgsql as
  $$
  declare
     numrep integer;
     nomerep varchar(50);
  begin
     select numeroreparto, nomereparto into numrep, nomerep from prodotto P
     where new.codprodotto = P.codice;

     if ((numrep != new.numeroreparto) or (nomerep != new.nomereparto)) then
        raise notice 'Prodotto venduto da un altro reparto: impossibile ordinarlo!';
        return null;
     end if;

     return new;
  end;
 $$;

create trigger controlla_ordini
  before insert or update on ordinereparto
  for each row
  execute procedure check_prodotti();
```

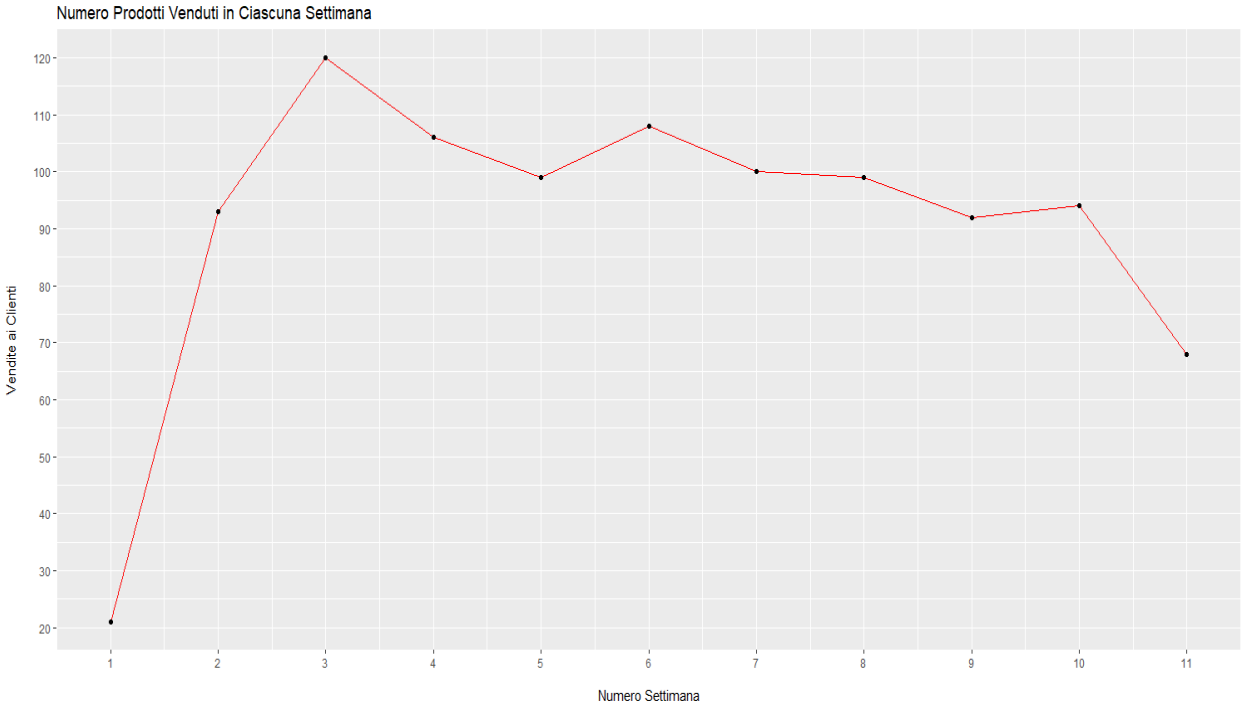
6 Analisi dei dati

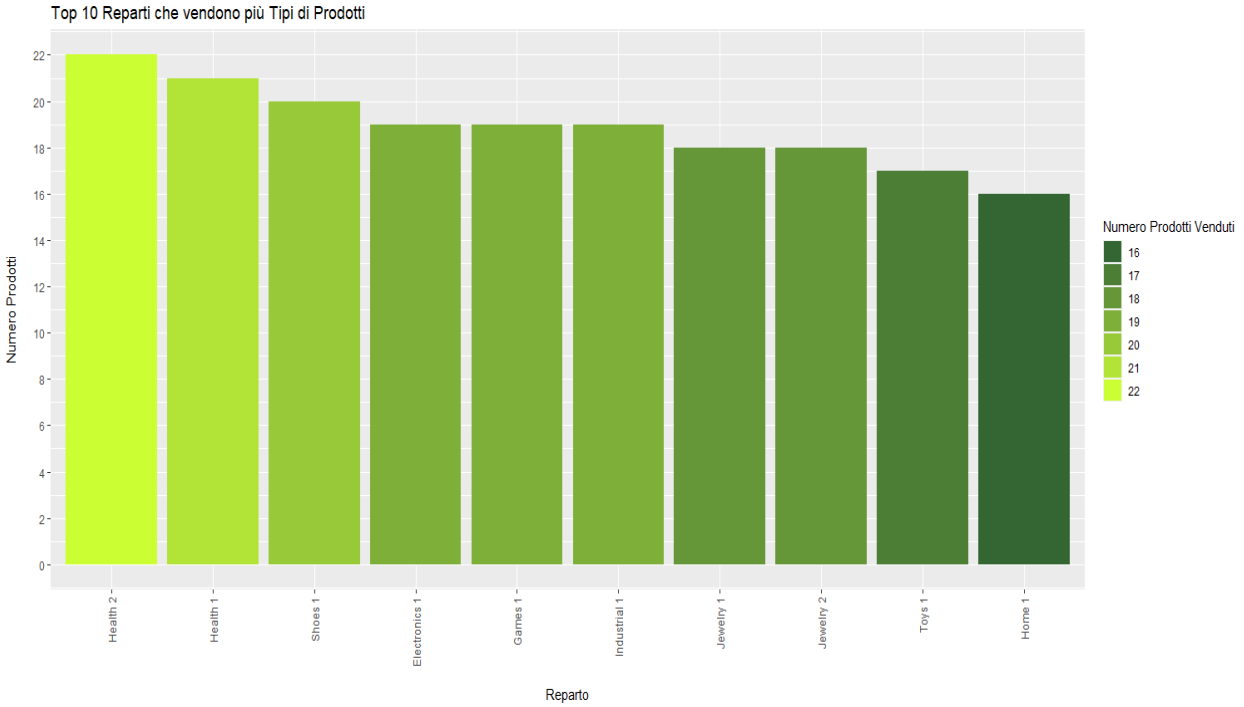





Distribuzione Dipendenti nei Reparti
Numero Dipendenti
0
1
2
3
4
5
6
7
8
9
10
Automotive 1
Beauty 1
Books 1
Books 2
Clothing 1
Electronics 1
Games 1
Games 2
Health 1
Health 2
Home 1
Industrial 1
Industrial 2
Jewelry 1
Jewelry 2
Shoes 1
Shoes 2
Shoes 3
Sports 1
Toys 1
Reparto
Reparto
1
3
5
7
9


Top 10 Prodotti più Venduti ai Clienti
Quantità venduta
1
2
3
4
5
6
7
8
9
115
141
277
1
44
243
300
18
55
62
CodiceProdotto
Quantità Venduta
7
8
9


Numero Prodotti Venduti in Ciascuna Settimana
Vendite ai Clienti
120
110
100
90
80
70
60
50
40
30
20
1
2
3
4
5
6
7
8
9
10
11
Numero Settimana


Top 10 Reparti che vendono più Tipi di Prodotti
Numero Prodotti
22
20
18
16
14
12
10
8
6
4
2
0
Health 2
Health 1
Shoes 1
Electronics 1
Games 1
Industrial 1
Jewelry 1
Jewelry 2
Toys 1
Home 1
Reparto
Numero Prodotti Venduti
16
17
18
19
20
21
22