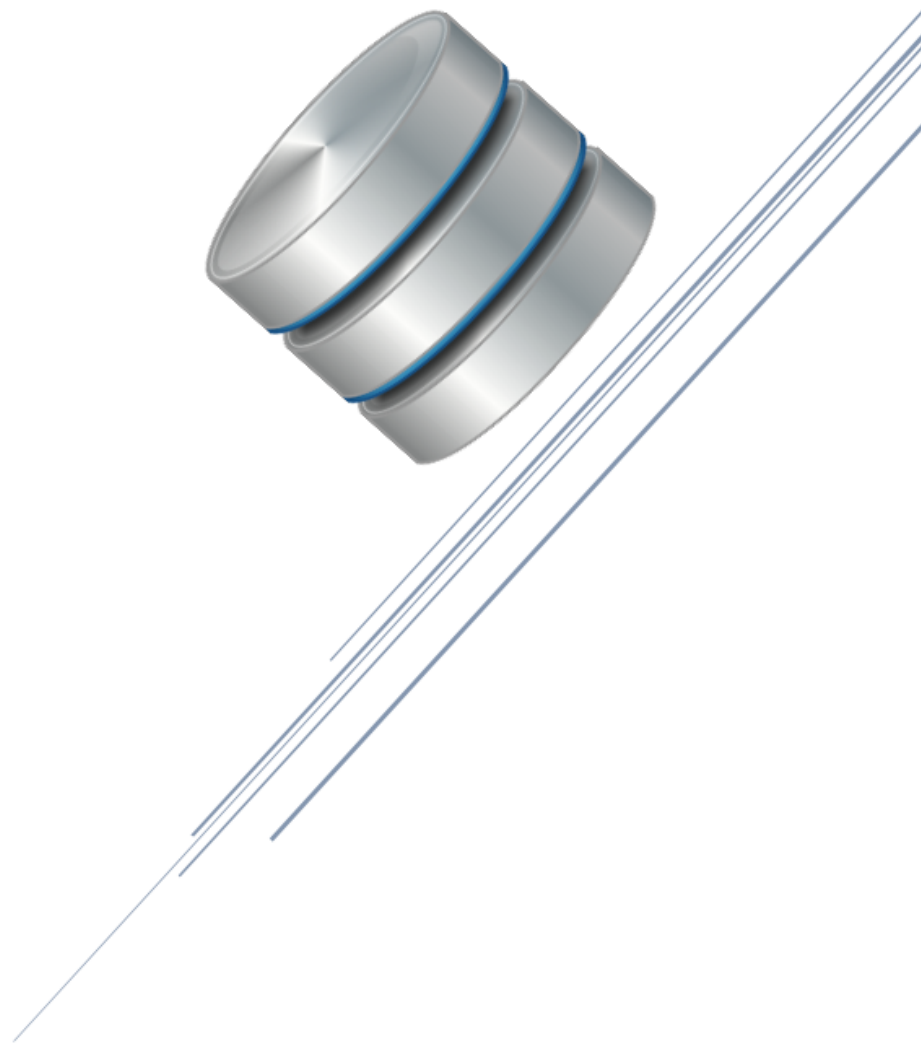


# DataBase

---

Stefano Nordio



Email: [stefanonordio96@gmail.com](mailto:stefanonordio96@gmail.com)  
LinkedIn: <https://www.linkedin.com/in/stefano-nordio>

## Indice

<b>1</b>	<b>Abstract</b>	<b>1</b>
<b>2</b>	<b>Analisi dei Requisiti</b>	<b>2</b>
<b>3</b>	<b>Progettazione Concettuale</b>	<b>5</b>
3.1	Entità e Attributi . . . . .	5
3.1.1	Dipendente . . . . .	5
3.1.2	Reparto . . . . .	5
3.1.3	Prodotto . . . . .	6
3.1.4	Fornitore . . . . .	6
3.1.5	Ordini . . . . .	6
3.1.6	Riparazioni . . . . .	6
3.1.7	Acquisto . . . . .	7
3.1.8	Fattura . . . . .	7
3.1.9	Cliente . . . . .	7
3.1.10	FidelityCard . . . . .	7
3.1.11	Pagamento . . . . .	7
3.2	Relazioni . . . . .	8
3.2.1	Dipendente – Reparto: Dirige . . . . .	8
3.2.2	Dipendente – Reparto: Lavora . . . . .	8
3.2.3	Prodotto – Reparto: Collocato . . . . .	8
3.2.4	Magazzino – Ordini: Inoltra . . . . .	8
3.2.5	Centro Riparazioni – Riparazioni: Effettua . . . . .	9
3.2.6	Prodotto – Fornitore: Fornito . . . . .	9
3.2.7	Prodotto – Acquisto: Riferito . . . . .	9
3.2.8	Acquisto – Pagamento: Comporta . . . . .	9
3.2.9	Acquisto – Fattura: Generato . . . . .	9
3.2.10	Acquisto – Cliente: Effettuato . . . . .	10
3.2.11	Cliente – FidelityCard: Possiede . . . . .	10
3.3	Diagramma ER . . . . .	11
<b>4</b>	<b>Progettazione Logica</b>	<b>12</b>
4.1	Valutazione di Costo . . . . .	12
4.1.1	Operazione . . . . .	12
4.1.2	Tavola dei Volumi . . . . .	12
4.1.3	Tavola degli Accessi . . . . .	13
4.1.4	Costo . . . . .	13
4.2	Analisi di una Ridondanza . . . . .	13
4.2.1	Premessa . . . . .	13
4.2.2	Operazione . . . . .	13
4.2.3	Tavola dei Volumi . . . . .	14
4.2.4	Tavola degli Accessi con Ridondanza (presenza dell'attributo "quantitàProd") . . . . .	14

4.2.5	Costo con Ridondanza . . . . .	14
4.2.6	Tavola degli Accessi senza Ridondanza (assenza dell'attributo "quantitàProd") . . . . .	14
4.2.7	Costo senza Ridondanza . . . . .	14
4.2.8	Conclusione . . . . .	14
4.3	Eliminazione delle Generalizzazioni . . . . .	15
4.3.1	Generalizzazione Tipi di Pagamento . . . . .	15
4.3.2	Generalizzazione Reparti . . . . .	15
4.4	Diagramma ER Ristrutturato . . . . .	16
4.5	Schema Logico . . . . .	17
4.6	Schema Relazionale . . . . .	18
<b>5</b>	<b>Query</b> . . . . .	<b>19</b>
5.1	Query 1 . . . . .	19
5.1.1	Operazione . . . . .	19
5.1.2	Codice . . . . .	19
5.1.3	Output . . . . .	19
5.2	Query 2 . . . . .	19
5.2.1	Operazione . . . . .	19
5.2.2	Codice . . . . .	19
5.2.3	Output . . . . .	20
5.3	Query 3 . . . . .	20
5.3.1	Operazione . . . . .	20
5.3.2	Codice . . . . .	20
5.3.3	Output . . . . .	20
5.4	Query 4 . . . . .	21
5.4.1	Operazione . . . . .	21
5.4.2	Codice . . . . .	21
5.4.3	Output . . . . .	21
5.5	Query 5 . . . . .	21
5.5.1	Operazione . . . . .	21
5.5.2	Codice . . . . .	21
5.5.3	Output . . . . .	22
5.6	Query 6 . . . . .	22
5.6.1	Operazione . . . . .	22
5.6.2	Codice . . . . .	22
5.6.3	Output . . . . .	23
<b>6</b>	<b>Funzioni</b> . . . . .	<b>24</b>
6.1	Funzione 1 . . . . .	24
6.1.1	Operazione . . . . .	24
6.1.2	Codice . . . . .	24
6.2	Funzione 2 . . . . .	24
6.2.1	Operazione . . . . .	24
6.2.2	Codice . . . . .	24

6.3	Funzione 3 . . . . .	25
6.3.1	Operazione . . . . .	25
6.3.2	Codice . . . . .	25
<b>7</b>	<b>Trigger</b>	<b>26</b>
7.1	Trigger 1 . . . . .	26
7.1.1	Operazione . . . . .	26
7.1.2	Codice . . . . .	26
7.2	Trigger 2 . . . . .	26
7.2.1	Operazione . . . . .	26
7.2.2	Codice . . . . .	26
7.3	Trigger 3 . . . . .	27
7.3.1	Operazione . . . . .	27
7.3.2	Codice . . . . .	27
7.4	Trigger 4 . . . . .	28
7.4.1	Operazione . . . . .	28
7.4.2	Codice . . . . .	28

## Elenco delle figure

1	Diagramma ER . . . . .	11
2	Diagramma Reparto-Prodotto-Fornitore . . . . .	12
3	Ridondanza . . . . .	13
4	Generalizzazione Tipi di Pagamento . . . . .	15
5	Generalizzazione Reparti . . . . .	15
6	Diagramma ER Ristrutturato . . . . .	16
7	Schema Logico . . . . .	17
8	Query 1 . . . . .	19
9	Query 2 . . . . .	20
10	Query 3 . . . . .	20
11	Query 4 . . . . .	21
12	Query 5 . . . . .	22
13	Query 6 . . . . .	23

## 1 Abstract

Si vuole realizzare una base di dati per l'organizzazione di un negozio di elettronica di consumo.

Il negozio è strutturato in vari reparti nei quali sono collocati diverse tipologie di prodotti appartenenti a diversi brand al fine di garantire una più ampia scelta al cliente.

L'ordine dei prodotti e la relativa gestione logistica sono garantiti dal magazzino del negozio che, grazie ai rapporti commerciali intrattenuti con diversi fornitori a livello internazionale, inoltra le richieste al momento del raggiungimento di una quantità minima prestabilita.

La responsabilità nella gestione dei reparti, dei prodotti e ovviamente della vendita spetta ai dipendenti, supervisionati da un capo reparto scelto in base all'esperienza lavorativa maturata in negozio.

Gli acquisti effettuati dai clienti generano ovviamente una fattura, prova indispensabile per poter inviare il prodotto nel centro riparazioni interno in caso di rottura o malfunzionamento. Ad ogni fattura generata viene associato un codice di transazione per il relativo tipo di pagamento scelto dal cliente.

In ogni caso, per premiare la fedeltà degli acquirenti, il negozio offre la possibilità di sottoscrivere gratuitamente una carta fedeltà che permette l'accumulo di punti ad ogni acquisto.

## 2 Analisi dei Requisiti

Si vuole realizzare una base di dati per la gestione e l'organizzazione di un negozio di elettronica, in particolare si vogliono memorizzare tutti i dati relativi ai prodotti collocati nei vari reparti, tutti i dati relativi al personale addetto al negozio e tutti i dati relativi agli acquisti effettuati dai clienti.

Il negozio è suddiviso nei reparti:

- Console;
- Telefonia;
- Audio;
- Informatica;
- Video;
- Elettrodomestici;
- Magazzino;
- Centro Riparazioni.

Ogni reparto è identificato da:

- un codice, che lo identifica univocamente;
- la quantità di prodotti collocati nel reparto.

Per ogni dipendente si vuole conoscere:

- un codice, per identificarlo univocamente;
- dati anagrafici (nome, cognome e data di nascita);
- dati lavorativi (data di assunzione e stipendio).

I prodotti disponibili nei vari reparti vengono memorizzati con le seguenti informazioni:

- un codice, differente per ogni prodotto;
- la marca;
- il nome;
- il colore;
- il prezzo;
- la quantità, disponibile nel negozio, necessaria al riordino del prodotto al momento del raggiungimento di una certa soglia prestabilita.

Gli ordini dei prodotti vengono effettuati dal magazzino e per ognuno di essi è importante tenere conto di:

- codice ordine, diverso per ogni ordine;
- prodotto;
- quantità;
- fornitore;
- data.

Il riordino dei prodotti viene inoltrato a vari fornitori dei quali si vuole conoscere:

- partita iva, che lo identifica univocamente;
- nome;
- sede (nazione, città, indirizzo).

Un acquisto può comprendere uno o più prodotti, ed è identificato da un codice acquisto univoco che genera una fattura, memorizzando:

- il numero della fattura, che conteggia il numero di fatture giornaliere;
- la data, che insieme al numero identifica univocamente la fattura;
- il totale.

Il centro riparazioni può ricevere prodotti acquistati e malfunzionanti e per ogni riparazione è necessario conoscere:

- il codice della riparazione, per identificarla in maniera esclusiva;
- il prodotto e il relativo codice d'acquisto per provarne l'effettivo acquisto al negozio;
- il prezzo della riparazione.

Un acquisto comporta un pagamento di cui vogliamo memorizzare l'id della transazione, univoco per ogni pagamento. Il pagamento relativo ai prodotti acquistati può essere effettuato in tre diverse modalità dai clienti:

- contanti;
- carte, specificando il circuito utilizzato;
- rateizzato, indicando il relativo numero di rate.

Gli acquisti vengono effettuati dai clienti dei quali vogliamo conoscere:

- codice fiscale, che li identificherà in maniera esclusiva;
- dati anagrafici (nome, cognome);



- contatti (mail, telefono).

Inoltre i clienti hanno la possibilità di avere una fidelity card, identificata da:

- numero fidelity, diverso per ogni carta;
- data di scadenza;
- punti accumulati.

## 3 Progettazione Concettuale

### 3.1 Entità e Attributi

#### 3.1.1 Dipendente

- **Descrizione:** modella un generico dipendente;
- **Attributi:**
  - codiceDip «PK» int
  - nome string
  - cognome string
  - dataNascita date
  - dataAssunzione date
  - stipendio decimal

#### 3.1.2 Reparto

- **Descrizione:** modella un generico reparto;
- **Attributi:**
  - codiceRep «PK» int

Sono definite le seguenti entità figlie di Reparto (generalizzazione di tipo totale):

1. **Console:** modella il reparto Console. Gli attributi che interessano del reparto Console sono gli stessi che per un reparto generico;
2. **Telefonia:** modella il reparto Telefonia. Gli attributi che interessano del reparto Telefonia sono gli stessi che per un reparto generico;
3. **Audio:** modella il reparto Audio. Gli attributi che interessano del reparto Audio sono gli stessi che per un reparto generico;
4. **Informatica:** modella il reparto Informatica. Gli attributi che interessano del reparto Informatica sono gli stessi che per un reparto generico;
5. **Video:** modella il reparto Video. Gli attributi che interessano del reparto Video sono gli stessi che per un reparto generico;
6. **Elettrodomestici:** modella il reparto Elettrodomestici. Gli attributi che interessano del reparto Elettrodomestici sono gli stessi che per un reparto generico;
7. **Magazzino:** modella il reparto Magazzino. Gli attributi che interessano del reparto Magazzino sono gli stessi che per un reparto generico;
8. **Centro Riparazioni:** modella il reparto Centro Riparazioni. Gli attributi che interessano del reparto Centro Riparazioni sono gli stessi che per un reparto generico.

### 3.1.3 Prodotto

- **Descrizione:** modella un generico prodotto;
- **Attributi:**
  - codiceProd «PK» int
  - marca string
  - nome string
  - colore string
  - prezzo decimal
  - quantità int

### 3.1.4 Fornitore

- **Descrizione:** modella un generico fornitore di prodotti;
- **Attributi:**
  - partitaIVA «PK» string
  - nome string
  - sede (attributo composto):
    - \* nazione string
    - \* città string
    - \* indirizzo string

### 3.1.5 Ordini

- **Descrizione:** modella un generico ordine inoltrato dal magazzino al fornitore;
- **Attributi:**
  - codiceOrd «PK» int
  - prodotto int
  - quantità int
  - fornitore string
  - dataOrd date

### 3.1.6 Riparazioni

- **Descrizione:** modella una generica riparazione ad un prodotto acquistato;
- **Attributi:**
  - codiceRip «PK» int
  - prodotto int
  - codiceAcq int
  - prezzo decimal

### 3.1.7 Acquisto

- **Descrizione:** modella un generico acquisto relativo a uno o più prodotti;
- **Attributi:**
  - codiceAcq «PK» int

### 3.1.8 Fattura

- **Descrizione:** modella una generica fattura relativa ad un acquisto;
- **Attributi:**
  - numeroFatt «PK» int
  - dataFatt «PK» date
  - totale decimal

### 3.1.9 Cliente

- **Descrizione:** modella un generico cliente;
- **Attributi:**
  - codFiscale «PK» string
  - nome string
  - cognome string
  - email string
  - telefono string

### 3.1.10 FidelityCard

- **Descrizione:** modella una generica carta fedeltà posseduta da un cliente;
- **Attributi:**
  - numeroCard «PK» int
  - punti int
  - dataScadenza date

### 3.1.11 Pagamento

- **Descrizione:** modella il tipo di pagamento utilizzato dal cliente per saldare l'acquisto;
- **Attributi:**
  - idTransazione «PK» int

Sono definite le seguenti entità figlie di Pagamento (generalizzazione di tipo totale):

1. **Contanti:** modella il tipo di pagamento in contanti. Gli attributi che interessano del tipo di pagamento in contanti sono gli stessi che per un tipo di pagamento generico;
2. **Carte:** modella il tipo di pagamento con carta;
  - **Attributi:** circuito string
3. **Rateizzato:** modella il tipo di pagamento a rate;
  - **Attributi:** numeroRate int

## 3.2 Relazioni

### 3.2.1 Dipendente – Reparto: Dirige

- Un dipendente può dirigere un solo reparto e un reparto è supervisionato da un solo dipendente;
- La molteplicità è dunque (1:1);
- L'associazione è parziale verso reparto e totale verso dipendente in quanto un dipendente potrebbe non dirigere nessun reparto mentre un reparto deve essere supervisionato da un dipendente.

### 3.2.2 Dipendente – Reparto: Lavora

- Un dipendente lavora in un solo reparto e in un reparto possono lavorare più dipendenti;
- La molteplicità è dunque (N:1);
- L'associazione è totale verso reparto e totale verso dipendente in quanto un dipendente deve lavorare in un reparto e in un reparto deve lavorare almeno un dipendente.

### 3.2.3 Prodotto – Reparto: Collocato

- Un prodotto è collocato al più in un reparto e in un reparto possono essere esposti più prodotti;
- La molteplicità è dunque (N:1);
- L'associazione è totale verso reparto e totale verso prodotto in quanto un prodotto deve essere collocato in un reparto e in un reparto viene esposto almeno un prodotto.

### 3.2.4 Magazzino – Ordini: Inoltra

- Il magazzino può inoltrare più ordini mentre un ordine è inoltrato dal magazzino;
- La molteplicità è dunque (1:N);
- L'associazione è parziale verso ordini e totale verso magazzino in quanto il magazzino potrebbe non aver ancora effettuato nessun ordine mentre un ordine è effettuato dal magazzino.

### 3.2.5 Centro Riparazioni – Riparazioni: Effettua

- Il centro riparazioni può effettuare più riparazioni mentre una riparazione è effettuata dal centro riparazioni;
- La molteplicità è dunque (1:N);
- L'associazione è parziale verso riparazioni e totale verso centro riparazioni in quanto il centro riparazioni può non aver effettuato nessuna riparazione mentre una riparazione è effettuata dal centro riparazioni.

### 3.2.6 Prodotto – Fornitore: Fornito

- Un prodotto può essere fornito da più fornitori e un fornitore può fornire diversi prodotti;
- La molteplicità è dunque (N:M);
- L'associazione è totale verso fornitore e totale verso prodotto in quanto un fornitore fornisce almeno un prodotto e un prodotto è fornito da almeno un fornitore.

### 3.2.7 Prodotto – Acquisto: Riferito

- Un prodotto può riferirsi a più acquisti e un acquisto può riguardare diversi prodotti;
- La molteplicità è dunque (N:M);
- L'associazione è parziale verso acquisto e totale verso prodotto in quanto un prodotto potrebbe non essere coinvolto in nessun acquisto mentre un acquisto coinvolge almeno un prodotto;
- **Attributi:** quantità int

### 3.2.8 Acquisto – Pagamento: Comporta

- Un acquisto comporta un pagamento e un pagamento è riferito ad un acquisto;
- La molteplicità è dunque (1:1);
- L'associazione è totale verso pagamento e totale verso acquisto in quanto un acquisto deve obbligatoriamente essere saldato da un pagamento e un pagamento si riferisce ad uno specifico acquisto.

### 3.2.9 Acquisto – Fattura: Generato

- Un acquisto genera una fattura e una fattura è generata all'acquisto;
- La molteplicità è dunque (1:1);
- L'associazione è totale verso fattura e totale verso acquisto in quanto un acquisto genera una e una sola fattura e una fattura si riferisce ad uno specifico acquisto.

### 3.2.10 Acquisto – Cliente: Effettuato

- Un acquisto è effettuato da un cliente mentre un cliente può effettuare più acquisti;
- La molteplicità è dunque (N:1);
- L'associazione è totale verso cliente e totale verso acquisto in quanto un acquisto è effettuato da un cliente e un cliente, in quanto tale, ha effettuato almeno un acquisto.

### 3.2.11 Cliente – FidelityCard: Possiede

- Un cliente può possedere al più una fidelity card e una fidelity card fa riferimento ad un cliente;
- La molteplicità è dunque (1:1);
- L'associazione è parziale verso fidelity card e totale verso cliente in quanto un cliente può non possedere la carta fedeltà mentre la fidelity card è intestata ad uno specifico cliente.

### 3.3 Diagramma ER

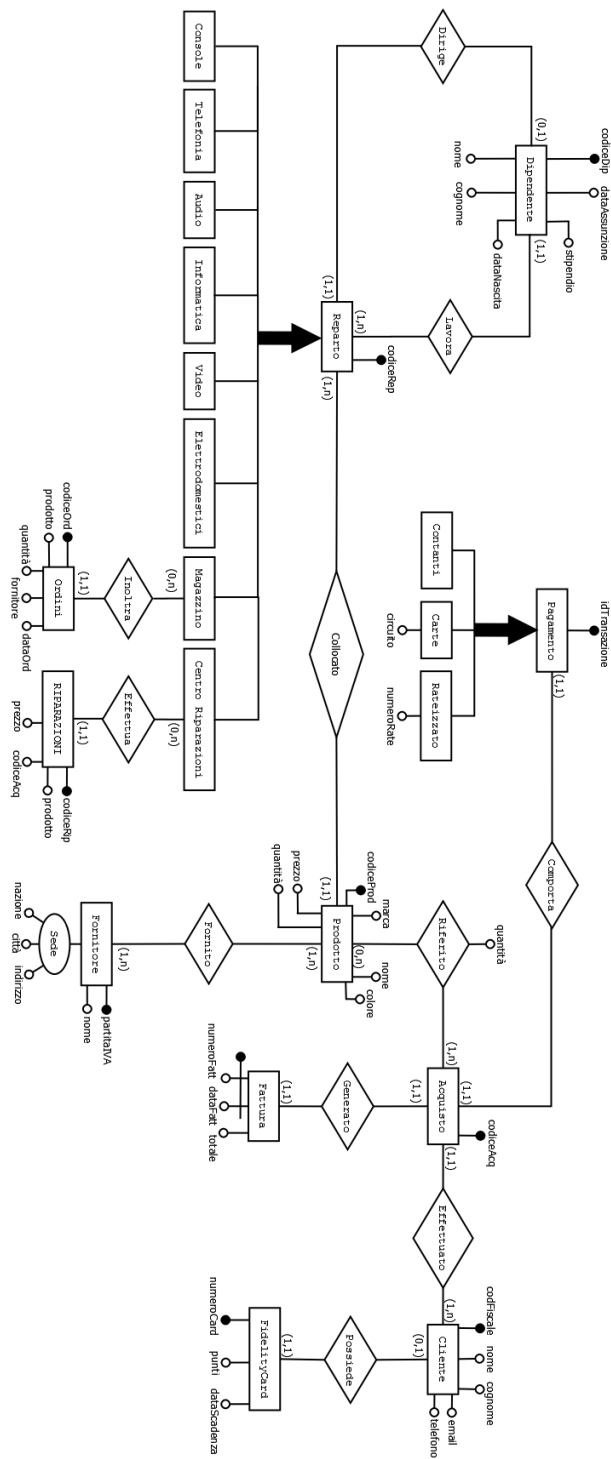


Figura 1: Diagramma ER



## 4 Progettazione Logica

### 4.1 Valutazione di Costo

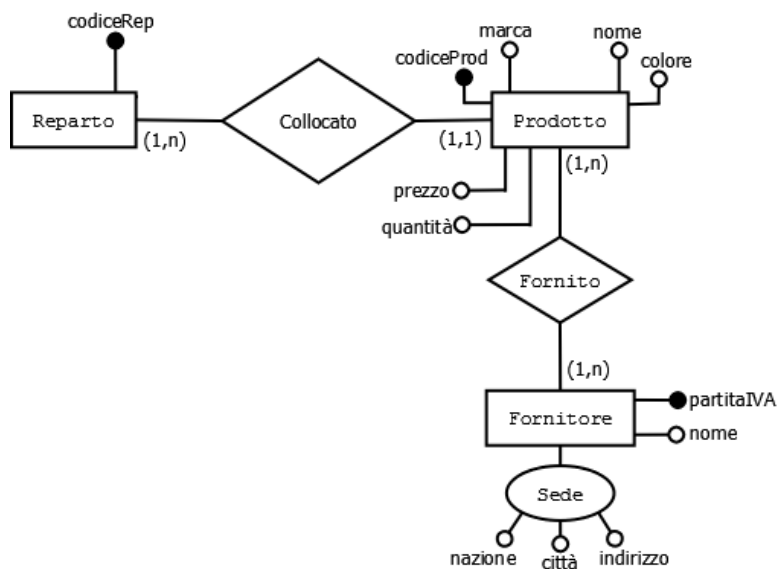


Figura 2: Diagramma Reparto-Prodotta-Fornitore

#### 4.1.1 Operazione

Trovare tutti i dati di un prodotto, del relativo fornitore e del reparto nel quale è collocato.

#### 4.1.2 Tavola dei Volumi

Concetto	Tipo	Volume
Prodotto	Entità	5000
Reparto	Entità	7
Fornitore	Entità	10
Fornito	Relazione	20000
Collocato	Relazione	5000

Tabella 1: Tabella dei Volumi

### 4.1.3 Tavola degli Accessi

Concetto	Costrutto	Accessi	Tipo
Prodotto	Entità	1	Lettura
Collocato	Relazione	1	Lettura
Reparto	Entità	1	Lettura
Fornito	Relazione	4	Lettura
Fornitore	Entità	4	Lettura

Tabella 2: Tabella degli Accessi

### 4.1.4 Costo

Per ricavare le informazioni di un prodotto e del relativo reparto nel quale è collocato si dovranno compiere una lettura di "Prodotto", una lettura di "Collocato" e una lettura di "Reparto" in quanto un prodotto può essere collocato in un solo reparto.

Per quanto riguarda invece i dati relativi ai fornitori bisogna tener conto della tavola dei volumi perché un prodotto può essere fornito da diversi fornitori. Perciò il numero di accessi in lettura nella relazione "Fornito" e nell'entità "Fornitore" sarà pari al rapporto del volume di "Fornito" col volume di "Prodotto", ossia  $\frac{20000}{5000} = 4$ .

## 4.2 Analisi di una Ridondanza

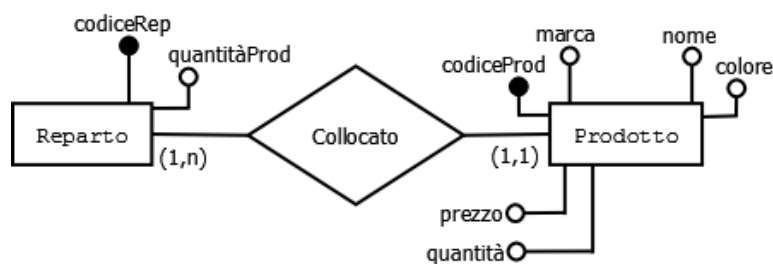


Figura 3: Ridondanza

### 4.2.1 Premessa

In tutte le operazioni e negli esempi in seguito, i reparti considerati sono 7 in quanto viene escluso il reparto centro riparazioni dove sono collocati prodotti non più disponibili alla vendita.

### 4.2.2 Operazione

Stampare la quantità totale di prodotti disponibili in un reparto (3 volte al giorno).

### 4.2.3 Tavola dei Volumi

Concetto	Tipo	Volume
Prodotto	Entità	5000
Reparto	Entità	7
Collocato	Relazione	5000

Tabella 3: Tabella dei Volumi

### 4.2.4 Tavola degli Accessi con Ridondanza (presenza dell'attributo "quantitàProd")

Concetto	Costrutto	Accessi	Tipo
Reparto	Entità	1	Lettura

Tabella 4: Tabella degli Accessi con Presenza di Ridondanza

### 4.2.5 Costo con Ridondanza

Un accesso in lettura a "Reparto" per ricavarsi la quantità di prodotti.

Il costo totale sarà quindi **3 accessi al giorno**.

### 4.2.6 Tavola degli Accessi senza Ridondanza (assenza dell'attributo "quantitàProd")

Concetto	Costrutto	Accessi	Tipo
Reparto	Entità	1	Lettura
Collocato	Relazione	715	Lettura

Tabella 5: Tabella degli Accessi con Assenza di Ridondanza

### 4.2.7 Costo senza Ridondanza

Per trovare i prodotti collocati in un reparto bisogna fare un accesso in lettura a "Reparto" e, seguendo la tavola degli accessi,  $\frac{5000}{7} = 715$  accessi in lettura a "Collocato".

Il costo totale sarà quindi **2145 accessi al giorno**.

### 4.2.8 Conclusione

In questa analisi di ridondanza risulta chiaro che aggiungere l'attributo "quantitàProd" nell'entità "Reparto" rende molto più efficiente e prestante il database in quanto, dovendo effettuare tre volte al giorno la stampa del numero di prodotti disponibili in un reparto,

mantenere la ridondanza significa compiere 3 accessi al giorno per ciascun reparto mentre rimuoverla comporta ben 2145 accessi al giorno per ciascun reparto.

### 4.3 Eliminazione delle Generalizzazioni

#### 4.3.1 Generalizzazione Tipi di Pagamento



Figura 4: Generalizzazione Tipi di Pagamento

Nella generalizzazione totale in "Pagamento" vengono eliminate le entità figlie e i loro attributi vengono aggiunti all'entità padre diventando opzionali, indicato dalla cardinalità (0,1). All'entità padre viene aggiunto un ulteriore attributo "tipo" per identificare le entità eliminate.

#### 4.3.2 Generalizzazione Reparti

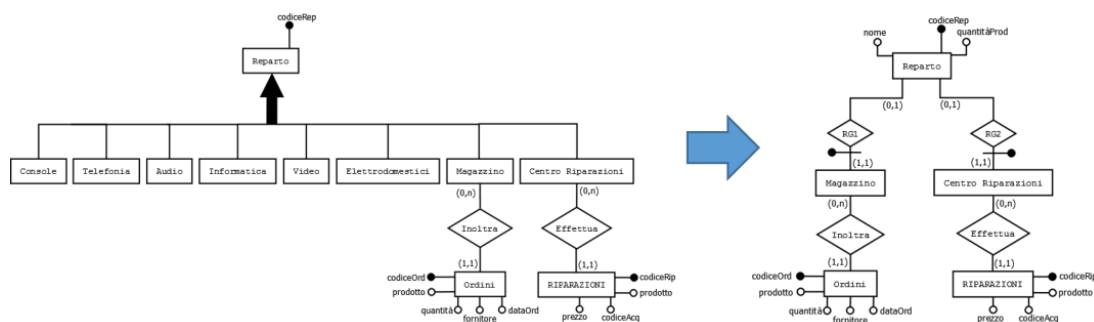


Figura 5: Generalizzazione Reparti

Nella generalizzazione totale in "Reparto" vengono mantenute solo le entità "Magazzino" e "Centro Riparazioni", le quali vengono messe in associazione con l'entità padre "Reparto" e sono identificate esternamente tramite l'associazione.

La cardinalità (0,1) indica che per tale associazione l'entità padre può avere zero o una entità figlio, mentre la cardinalità (1,1) indica che l'entità figlio può avere un solo padre.

Le restanti entità figlie vengono eliminate aggiungendo il nuovo attributo "nome" per identificarle.

#### 4.4 Diagramma ER Ristrutturato

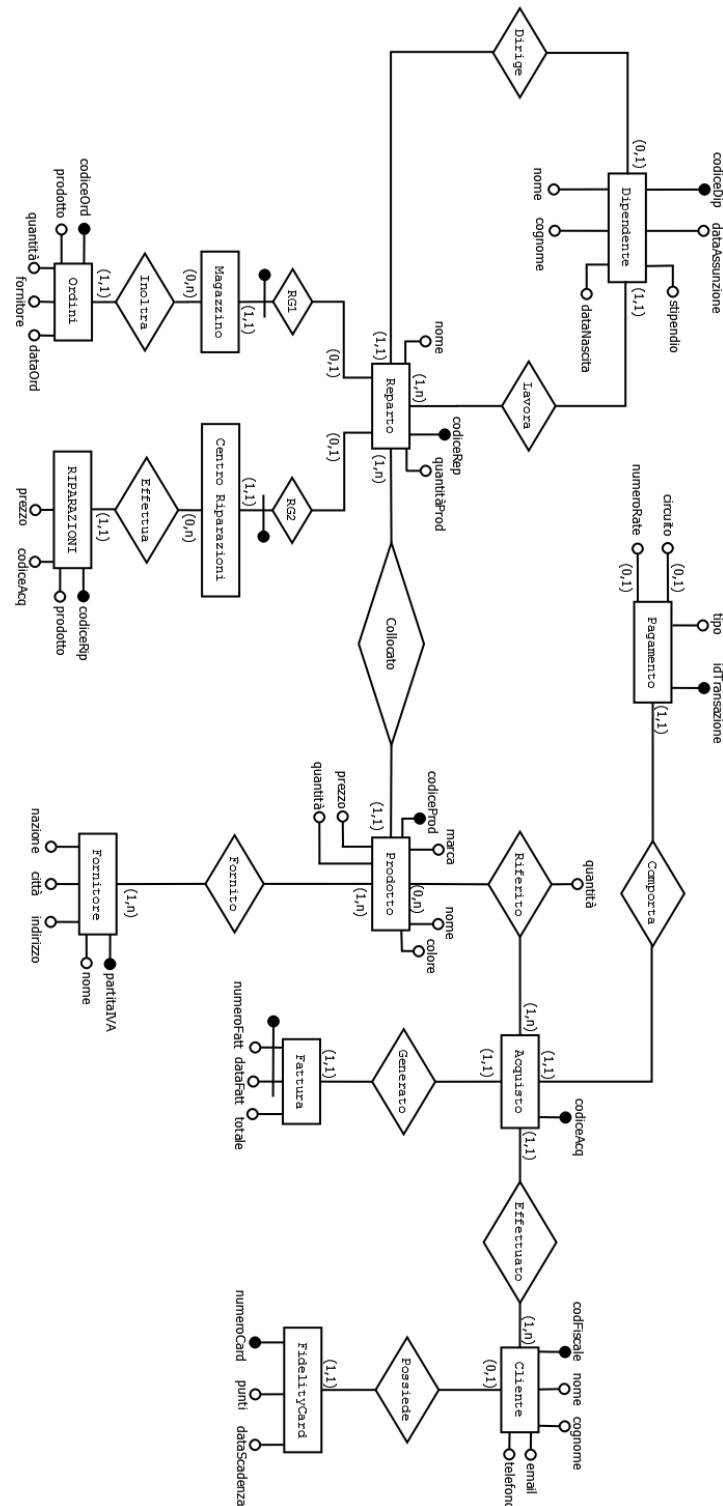


Figura 6: Diagramma ER Ristrutturato

## 4.5 Schema Logico



#### 4.6 Schema Relazionale

1. DIPENDENTE (codiceDip, nome, cognome, dataNascita, dataAssunzione, stipendio, reparto)
2. REPARTO (codiceRep, nome, capoRep, quantitàProd)
3. PRODOTTO (codiceProd, reparto, marca, nome, colore, prezzo, quantità)
4. FORNITURAPRODOTTI (prodotto, partitaIVA)
5. FORNITORE (partitaIVA, nome, nazione, città, indirizzo)
6. ORDINI (reparto, codiceOrd, prodotto, quantità, fornitore, dataOrd)
7. RIPARAZIONI (reparto, codiceRip, prodotto, codiceAcq, prezzo)
8. ACQUISTIPRODOTTI (prodotto, acquisto, quantità)
9. ACQUISTO (codiceAcq, cliente)
10. CLIENTE (codFiscale, nome, cognome, email, telefono)
11. FIDELITY CARD (numeroCard, punti, dataScadenza, cliente)
12. FATTURA (numeroFatt, dataFatt, totale, acquisto)
13. PAGAMENTO (idTransazione, acquisto, tipo, circuito, numeroRate)

**Osservazione:** Le entità "Magazzino" e "Centro Riparazioni" che erano state mantenute nella ristrutturazione dello schema ER, sono state inglobate rispettivamente nelle relazioni "Ordini" e "Riparazioni" aggiungendo l'attributo reparto, con vincolo di integrità referenziale verso la chiave di "Reparto", che dovrà sempre essere impostato per default al valore "7" nella relazione "Ordini" (in quanto 7 è il codice identificativo del reparto "Magazzino") e "8" nella relazione "Riparazioni" (in quanto 8 è il codice identificativo del reparto "Magazzino").

## 5 Query

### 5.1 Query 1

#### 5.1.1 Operazione

Mostrare per gli articoli forniti dal fornitore "Klijesta", la marca, il nome, il prezzo di vendita e la quantità disponibile in negozio (comprendente la quantità disponibile in reparto + la quantità a magazzino).

#### 5.1.2 Codice

```
SELECT p.marca, p.nome, p.prezzo, SUM(p.quantita) AS quantitaDisp
FROM Prodotto p
WHERE p.codiceProd IN(
    SELECT fp.prodotto
    FROM FornituraProdotti fp JOIN Fornitore f ON f.partitaIVA=fp.partitaIVA
    WHERE f.nome='Klijesta')
GROUP BY p.codiceProd;
```

#### 5.1.3 Output

marca	nome	prezzo	quantitaDisp
Apple	MacBook Pro 15	2799.00	14
Sony	Playstation 4	389.99	18
JBL	Go	29.99	44

Figura 8: Query 1

### 5.2 Query 2

#### 5.2.1 Operazione

Mostrare per i reparti aventi almeno 4 dipendenti, il nome del reparto, lo stipendio minimo, quello massimo e quello medio dei relativi dipendenti (troncandolo a 2 cifre dopo la virgola).

#### 5.2.2 Codice

```
SELECT r.nome AS reparto, MIN(d.stipendio) AS stipendioMIN, MAX(d.stipendio)
    AS stipendioMAX, CAST(AVG(d.stipendio) AS decimal(7,2)) AS stipendioMEDIO
FROM Reparto r JOIN Dipendente d ON r.codiceRep=d.reparto
GROUP BY r.nome
HAVING COUNT(r.nome)>=4;
```



### 5.2.3 Output

reparto	stipendioMIN	stipendioMAX	stipendioMEDIO
Audio	1263.87	1650.72	1417.66
Console	1321.61	1597.77	1450.22
Elettrodomestici	1349.03	1700.25	1455.37
Informatica	1247.55	1708.30	1402.03
Telefonia	1297.54	1669.34	1460.72

Figura 9: Query 2

## 5.3 Query 3

### 5.3.1 Operazione

Mostrare nome e cognome dei clienti che nel mese di marzo 2017 hanno effettuato un acquisto superiore ai 2500 euro utilizzando come metodo di acquisto quello "Rateizzato". Mostrare anche il relativo numero di rate scelto dal cliente.

### 5.3.2 Codice

```
SELECT c.nome, c.cognome, p.numeroRate
FROM Cliente c, Pagamento p, Acquisto a
WHERE c.codFiscale=a.cliente AND a.codiceAcq=p.acquisto AND c.codFiscale IN(
    SELECT a1.cliente
    FROM Acquisto a1
    WHERE a1.codiceAcq IN(
        SELECT p1.acquisto
        FROM Fattura f JOIN Pagamento p1 ON f.totale>2500
        WHERE p1.tipo='Rateizzato' AND f.dataFatt BETWEEN '2017-03-01'
        AND '2017-03-31'));
```

### 5.3.3 Output

nome	cognome	numeroRate
Roberta	Ferrari	24
Anita	Gallo	24
Alessio	Castiglione	18
Rosita	Trentino	12

Figura 10: Query 3

## 5.4 Query 4

### 5.4.1 Operazione

Mostrare partitalVA, nome e città dei fornitori italiani che tra il 2015 e il 2017 abbiano fornito almeno 30 fra prodotti Apple e Samsung. Mostrare quindi anche la quantità totale fornita in ordine crescente.

### 5.4.2 Codice

```
SELECT f.partitalVA, f.nome, f.città, SUM(o.quantita) AS quantitaFornita
FROM Fornitore f JOIN Ordini o ON f.nazione='Italia' AND f.partitalVA=o.fornitore
WHERE o.dataOrd BETWEEN '2015-01-01' AND '2017-12-31' AND o.prodotto IN(
    SELECT p.codiceProd
    FROM Prodotto p
    WHERE p.marca='Apple' OR p.marca='Samsung')
GROUP BY f.partitalVA
ORDER BY quantitaFornita ASC;
```

### 5.4.3 Output

partitalVA	nome	città	quantitaFornita
IT13992538812	Piramide	Milano	35
IT24825483622	Esagono	Roma	95

Figura 11: Query 4

## 5.5 Query 5

### 5.5.1 Operazione

Mostrare il nome dei reparti che nel giorno 10 marzo 2017 hanno venduto almeno 3 prodotti. Siccome interessano le vendite, i reparti magazzino e centro riparazioni non devono venire considerati. Ordinare i risultati in modo decrescente per il numero di vendite di ogni reparto.

### 5.5.2 Codice

```
DROP VIEW IF EXISTS Acquisti10marzo;
CREATE VIEW Acquisti10marzo
AS SELECT ap.prodotto, SUM(ap.quantita) AS numVendite
FROM AcquistiProdotti ap JOIN Fattura f ON ap.acquisto=f.acquisto
WHERE f.dataFatt='2017-03-10'
GROUP BY ap.prodotto;

SELECT r.nome, am.numVendite
```

```
FROM Reparto r JOIN Prodotto p ON p.reparto<=6
  JOIN Acquisti10marzo am ON p.codiceProd=am.prodotto
WHERE p.reparto=r.codiceRep AND am.numVendite>=3
ORDER BY am.numVendite DESC;
```

### 5.5.3 Output

nome	numVendite
Audio	8
Elettrodomestici	5
Informatica	3

Figura 12: Query 5

## 5.6 Query 6

### 5.6.1 Operazione

Mostrare per i clienti aventi almeno 1000 punti sulla fidelity card, nome e cognome del cliente, marca e nome del prodotto e il relativo prezzo della riparazione per eventuali prodotti acquistati e inviati al centro riparazioni.

### 5.6.2 Codice

```
DROP VIEW IF EXISTS AcquistiClientiFedeli;
CREATE VIEW AcquistiClientiFedeli
AS SELECT c.nome, c.cognome, a.codiceAcq
FROM Cliente c JOIN Acquisto a ON c.codFiscale=a.cliente
WHERE a.cliente IN(
  SELECT fc.cliente
  FROM FidelityCard fc
  WHERE fc.punti>1000);
```

```
SELECT acf.nome, acf.cognome, p.marca AS marcaProd, p.nome AS nomeProd,
  r.prezzo AS prezzoRip
FROM AcquistiClientiFedeli acf JOIN Prodotto p JOIN Riparazioni r
  ON acf.codiceAcq=r.codiceAcq AND r.prodotto=p.codiceProd
GROUP BY r.codiceRip
ORDER BY acf.nome, acf.cognome;
```

## 5.6.3 Output

nome	cognome	marcaProd	nomeProd	prezzoRip
Alessio	Castiglione	Samsung	Galaxy S7	180.00
Anita	Gallo	JBL	Go	10.99
Lino	Trevisan	Apple	MacBook Pro 15	499.99
Roberta	Ferrari	Apple	MacBook Pro 15	250.99
Roberta	Ferrari	Samsung	Galaxy S7	180.00

Figura 13: Query 6

## 6 Funzioni

### 6.1 Funzione 1

#### 6.1.1 Operazione

La seguente funzione, dato in input il "codDip" di un certo dipendente, calcola il numero di scatti d'anzianità maturati dalla data d'assunzione in negozio (ipotizziamo che tutti i dipendenti abbiano un contratto che prevede massimo 10 scatti d'anzianità che maturano ogni 3 anni e ad ogni scatto corrisponde un aumento dello stipendio di euro 20,66).

#### 6.1.2 Codice

```
DROP FUNCTION IF EXISTS ScattiAnzianita;
DELIMITER $
CREATE FUNCTION ScattiAnzianita (codDip INT)
RETURNS INT(2)
BEGIN
DECLARE NumeroScatti INT(2);
SELECT FLOOR(DATEDIFF(CURDATE(), d.dataAssunzione)/365/3) INTO NumeroScatti
FROM Dipendente d
WHERE codiceDip=codDip;
IF(NumeroScatti>10)
THEN
SET NumeroScatti=10;
END IF;
RETURN NumeroScatti;
END $
DELIMITER ;
```

### 6.2 Funzione 2

#### 6.2.1 Operazione

La seguente funzione, dato in input il "codDip" di un certo dipendente, calcola la relativa retribuzione dovuta dall'anzianità di servizio (le ipotesi sono le medesime della funzione precedente).

#### 6.2.2 Codice

```
DROP FUNCTION IF EXISTS RetribuzioneAnzianita;
DELIMITER $
CREATE FUNCTION RetribuzioneAnzianita (codDip INT)
RETURNS DECIMAL(5,2)
BEGIN
DECLARE BonusAnzianita DECIMAL(5,2);
```

```
SELECT ScattiAnzianita(codDip)*20.66 INTO BonusAnzianita
FROM Dipendente d
WHERE codiceDip=codDip;
RETURN BonusAnzianita;
END $
DELIMITER ;
```

## 6.3 Funzione 3

### 6.3.1 Operazione

La seguente funzione, dato in input il "codAcq" di un certo acquisto effettuato scegliendo il metodo di pagamento rateizzato, restituisce il valore di ogni singola rata compresa degli interessi (ipotizziamo che il negozio offra la possibilità di finanziamento con tasso d'interesse annuo al 7% e con una spesa di incasso rata pari a euro 1,50). Per il calcolo della rata bisogna usare la seguente formula:  $\frac{C \times R}{\left(1 - \frac{1}{(1+R)^N}\right)}$  dove C è l'importo da finanziare, R è il tasso d'interesse mensile ( $\frac{0.07}{12} = 0.0058$ ) ed N è il numero delle rate. Ad ogni rata va poi sommato 1.50 euro per le spese di incasso rata.

### 6.3.2 Codice

```
DROP FUNCTION IF EXISTS CostoRata;
DELIMITER $
CREATE FUNCTION CostoRata (codAcq INT)
RETURNS DECIMAL(5,2)
BEGIN
DECLARE Rata Decimal(5,2);
SELECT f.totale*0.0058/(1-1/POWER(1+0.0058, p.numeroRate))+1.50 INTO Rata
FROM Pagamento p JOIN Fattura f ON p.acquisto=f.acquisto
WHERE p.acquisto=codAcq;
RETURN Rata;
END $
DELIMITER ;
```

## 7 Trigger

### 7.1 Trigger 1

#### 7.1.1 Operazione

Trigger che assegna alla tabella "reparto" il nuovo capo reparto nel caso in cui il precedente venga rimosso dalla tabella "dipendente" per licenziamento/pensionamento. Il nuovo capo reparto assegnato sarà, tra i dipendenti rimasti in quel reparto, quello con maggior esperienza lavorativa.

#### 7.1.2 Codice

```
DROP TRIGGER IF EXISTS NuovoCapoRep;
DELIMITER $
CREATE TRIGGER NuovoCapoRep
BEFORE DELETE ON Dipendente
FOR EACH ROW
BEGIN
    IF(SELECT r.capoRep
       FROM Reparto r JOIN Dipendente d ON r.capoRep=d.codiceDip
       WHERE r.capoRep=OLD.codiceDip)
    THEN
        UPDATE reparto
        SET capoRep=(SELECT codicedip
                     FROM Dipendente
                     WHERE reparto=OLD.reparto AND dataAssunzione=(SELECT
                        MIN(dataAssunzione)
                        FROM Dipendente
                        WHERE reparto=OLD.reparto AND dataAssunzione!=OLD.dataAssunzione))
        WHERE capoRep=OLD.codiceDip;
    END IF;
END $
DELIMITER ;
```

### 7.2 Trigger 2

#### 7.2.1 Operazione

Trigger che permette di gestire il conteggio del numero di fattura giornaliero reimpostando tale valore a 0 nel momento in cui inizia un nuovo giorno lavorativo.

#### 7.2.2 Codice

```
DROP TRIGGER IF EXISTS NumeroFattura;
DELIMITER $
```

```
CREATE TRIGGER NumeroFattura
BEFORE INSERT ON Fattura
FOR EACH ROW
BEGIN
DECLARE conteggio INTEGER;
DECLARE dataRecente DATE;
SELECT COUNT(*) FROM Fattura INTO conteggio;
SELECT MAX(dataFatt) FROM Fattura INTO dataRecente;
IF(conteggio=0)
THEN
SET NEW.numeroFatt=1;
END IF;
IF(conteggio>0)
THEN
    IF(NEW.dataFatt>dataRecente)
    THEN
        SET NEW.numeroFatt=1;
    ELSE IF(NEW.dataFatt=dataRecente)
    THEN
        SET NEW.numeroFatt=(SELECT MAX(numeroFatt) FROM Fattura
        WHERE dataFatt=dataRecente)+1;
    END IF;
END IF;
END IF;
END $
DELIMITER ;
```

### 7.3 Trigger 3

#### 7.3.1 Operazione

Trigger che effettua automaticamente il riordino di 50 pezzi di un prodotto non appena la sua quantità a magazzino scende al di sotto di 10 pezzi, inserendo il nuovo ordine nella tabella "ordini". Il nuovo ordine inserito viene inoltrato ad un fornitore scelto casualmente tra quelli che forniscono il prodotto da ordinare.

#### 7.3.2 Codice

```
DROP TRIGGER IF EXISTS Riordino50Pezzi;
DELIMITER $
CREATE TRIGGER Riordino50Pezzi
AFTER UPDATE ON Prodotto
FOR EACH ROW
BEGIN
DECLARE fornitore VARCHAR(13);
SELECT partitalVA
```



```
FROM FornituraProdotti
WHERE prodotto=NEW.codiceProd
ORDER BY RAND() LIMIT 1 INTO fornitore;
IF(OLD.quantita>=10)
THEN
    IF(NEW.reparto=7 AND NEW.quantita<10)
    THEN
        INSERT INTO 'Ordini' ('prodotto', 'quantita', 'fornitore', 'dataOrd')
        VALUES (NEW.codiceProd, 50, fornitore, CURDATE());
    END IF;
END IF;
END $
DELIMITER ;
```

## 7.4 Trigger 4

### 7.4.1 Operazione

Trigger che aggiorna i punti della fidelity card di un cliente, nel caso in cui ne sia in possesso, non appena una fattura viene generata a suo nome (ipotizziamo che 1 euro di spesa equivalga ad 1 punto sulla fidelity card).

### 7.4.2 Codice

```
DROP TRIGGER IF EXISTS AggiornoPunti;
DELIMITER $
CREATE TRIGGER AggiornoPunti
AFTER INSERT ON Fattura
FOR EACH ROW
BEGIN
    DECLARE puntiAcq INT(5);
    DECLARE clienteAcq CHAR(16);
    SELECT FLOOR(NEW.totale) INTO puntiAcq;
    SELECT cliente FROM Acquisto WHERE codiceAcq=NEW.acquisto INTO clienteAcq;
    UPDATE FidelityCard
    SET punti=punti+puntiAcq
    WHERE cliente=clienteAcq;
END $
DELIMITER ;
```