

HH: Hardware House

Cesare Omodei, Francesco De Marchi

AA: 2019/2020

Contents

1 Abstract	1
2 Analisi dei Requisiti	1
2.1 Descrizione testuale dei requisiti e operazioni tipiche	1
2.2 Glossario dei Termini	2
2.3 Frasi relative	2
3 Progettazione concettuale	4
3.1 Descrizione entità e relazioni	4
3.1.1 Entità ed attributi	4
3.1.2 Relazioni	5
3.2 Schema concettuale (E-R)	7
4 Progettazione logica	7
4.1 Ristrutturazione dello schema	7
4.1.1 Analisi delle ridondanze	7
4.1.2 Politiche di ristrutturazione dello schema	9
5 Schema Relazionale	10
5.1 Tabelle	10
5.2 Chiavi esterne	11
6 Query e Indici	11

6.1 Query	11
6.2 Indici	15

1 Abstract

Hardware House è il nome di una presunta catena di negozi situata nel padovano che nasce con l'obiettivo di prestare un servizio di selezione, assemblaggio e vendita di componenti hardware per pc.

HH nasce con l'obiettivo di aiutare gli utenti a navigare il mondo della componentistica per PC. La catena si offre quindi di dare una consulenza completamente gratuita ai clienti, accompagnandoli nell'intero processo di scelta e assemblaggio dei componenti che comporranno il loro computer, in base alle caratteristiche tecniche ricercate e alle diverse funzioni che esso dovrà svolgere.

Inoltre la catena dispone anche di pc pre-assemblati con configurazioni di diverse fasce di prezzo e funzionalità e mette a disposizione degli esperti hardware per la richiesta di riparazione o assemblaggio dei componenti.

Il progetto offre un' implementazione della base di dati relativa alla catena di negozi, preoccupandosi della sua gestione dal punto di vista organizzativo e amministrativo. Si preoccupa dunque di costruire le fondamenta necessarie allo sviluppo delle varie funzioni che Hardware House fornisce ai clienti.

Funzioni fondamentali quali la gestione di un negozio, la parte organizzativa e di supporto ai clienti.

2 Analisi dei Requisiti

2.1 Descrizione testuale dei requisiti e operazioni tipiche

Lo scopo del progetto è quello di realizzare una base di dati che raccolga e permetta l'elaborazione di informazioni atte alla gestione efficace dei negozi della catena.

Per fare ciò si vogliono indicare le informazioni relative al negozio, a partire dal personale, dai componenti presenti in magazzino e le informazioni necessarie alla vendita di pc pre- assemblati e/o di componenti e per la riparazione di quest'ultimi.

Le informazioni necessarie alla rappresentazione di un **negozio** sono un ID che identifica univocamente un negozio, un indirizzo che indica la locazione del negozio e una città che presenta il nome delle città in cui è situato il negozio.

Ogni negozio ha un **magazzino** dove mantiene i componenti pronti alla vendita, i magazzini sono caratterizzati da un ID, da un indirizzo e da una città. Inoltre come informazione aggiuntiva si presenta il volume in metri cubi del magazzino.

I **componenti** hardware che sono presenti o meno nel magazzino sono identificati anche loro da un ID, dal nome del modello e dal prezzo associato ad essi. Si suddividono ovviamente in diversi tipi quali schede madri, CPU, schede video, sistemi di raffreddamento CPU, RAM, memorie, case e alimentatori.

I componenti sono inoltre associati alle **case produttrici** che li fabbricano, di cui si mantengono le informazioni relative al nome della compagnia, all'email e al contatto telefonico.

I **PC pre-assemblati** sono caratterizzati invece da un nome univoco, dal prezzo e dalla funzione principale che il pc con quella determinata configurazione sarà atto a fare.

Ad ogni negozio sono poi associati una serie di **dipendenti** che si suddividono in commessi, manutentori e consulenti. Essi sono identificati da un codice fiscale, dal nome, cognome, da uno stipendio e da una data di assunzione.

I commessi e i consulenti sono a diretto contatto con i **clienti**, che hanno l'obbligo di registrarsi per usufruire dei servizi dati dal negozio. Nella base di dati si mantengono le informazioni relative a quest'ultimi riportandone nome, cognome, un' email (se presente) per informarli di offerte pubblicitarie e un ID univoco.

Ogni cliente può o meno effettuare degli acquisti, che verranno registrati facendo riferimento al negozio in cui sono stati fatti. Gli **acquisti** avranno dunque un ID, una data che si riferisce a quando è stato effettuato l'acquisto e un totale che indica il prezzo totale della merce acquistata. Gli acquisti si possono classificare in acquisti di componenti e acquisti di pc pre-assemblati, i quali faranno riferimento relativamente ai componenti e ai pc. Gli acquisti di componenti a loro volta possono essere conclusi o non, in base alla presenza o meno dei componenti desiderati in magazzino. In quest'ultimo caso il negozio provvederà ad effettuare un ordine dei componenti mancanti in relazione all'acquisto non concluso. In ogni caso l'acquisto viene registrato perchè il cliente paga comunque al momento dell'ordine (se l'ordine non dovesse andare a buon fine il cliente avrà diritto ad un rimborso).

Un **ordine** è identificato dall'ID dell'acquisto non concluso, da una data in cui l'ordine è stato fatto e da una di arrivo previsto dell'ordine.

Infine il negozio ha la necessità di raccogliere e mantenere le informazioni necessarie alla **manutenzione** dei componenti, e ne riporta quindi la data della avvenuta manutenzione e il tipo di manutenzione (riparazione o assemblaggio di componenti). Nella manutenzione viene tenuto conto poi della durata di tale manutenzione, al fine di retribuire i lavoratori con un supplemento a ore oltre allo stipendio fisso mensile.

2.2 Glossario dei Termini

Termine	Significato	Sinonimo/i
Negozio	Sedi in cui si svolgono le mansioni previste da HH	Attività commerciale, sede
Magazzino	Stabilimento dove si situano i componenti	Deposito, stock, scorte
Componente	Pezzo di hardware che compone il pc	Parte, pezzo
Produttore	Compagnia che produce e fornisce i componenti	Realizzatore, fabbricante
Dipendente	Persona che svolge una mansione in un determinato negozio	Impiegato, lavoratore
Cliente	Individuo che beneficia dei servizi della catena	Consumatore, acquirente
Acquisto	Registrazione dell'avvenuta compravendita con il cliente	Compravendita, cessione
Ordine	Monitoraggio delle richieste di rifornimenti da parte dei produttori	Richiesta, commissione
Manutenzione	Operazione di assemblaggio e riparazione componenti	Mantenimento

2.3 Frasi relative

FRASI RELATIVE A NEGOZIO

Le informazioni necessarie alla rappresentazione di un **negozio** sono un ID che identifica univocamente un negozio, un indirizzo che indica la locazione del negozio e una città che presenta il nome delle città in cui è situato il negozio. Ogni negozio ha un **magazzino** dove mantiene i componenti pronti alla vendita

FRASI RELATIVE A MAGAZZINO

I magazzini sono caratterizzati da un ID, da un indirizzo e da una città. Inoltre come informazione aggiuntiva si presenta il volume in metri cubi del magazzino.

FRASI RELATIVE A COMPONENTE

I **componenti** hardware che sono presenti o meno nel magazzino sono identificati anche loro da un ID, dal nome del modello e dal prezzo associato ad essi. Si suddividono ovviamente in diversi tipi quali schede madri, CPU, schede video, sistemi di raffreddamento CPU, RAM, memorie, case e alimentatori.

FRASI RELATIVE A PRODUTTORE

I componenti sono inoltre associati alle **case produttrici** che li fabbricano, di cui si mantengono le informazioni relative al nome della compagnia, all'email e al contatto telefonico.

FRASI RELATIVE A PC

I **PC pre-assemblati** sono caratterizzati invece da un nome univoco, dal prezzo e dalla funzione principale che il pc con quella determinata configurazione sarà atto a fare.

FRASI RELATIVE A DIPENDENTE

Ad ogni negozio sono poi associati una serie di **dipendenti** che si suddividono in commessi, manutentori e consulenti. Essi sono identificati da un codice fiscale, dal nome, cognome, da uno stipendio e da una data di assunzione.

FRASI RELATIVE A CLIENTE

Nella base di dati si mantengono le informazioni relative a quest'ultimi riportandone nome, cognome, un' email per informarli di offerte pubblicitarie e un ID univoco. Ogni cliente può o meno effettuare degli acquisti, che verranno registrati facendo riferimento al negozio in cui sono stati fatti.

FRASI RELATIVE AD ACQUISTO

Gli **acquisti** avranno dunque un ID, una data che si riferisce a quando è stato effettuato l'acquisto e un totale che indica il prezzo totale della merce acquistata. Gli acquisti si possono classificare in acquisti di componenti e acquisti di pc pre-assemblati, i quali faranno riferimento relativamente ai componenti e ai pc. Gli acquisti di componenti a loro volta possono essere conclusi o non, in base alla presenza o meno dei componenti desiderati in magazzino. In quest'ultimo caso il negozio provvederà ad effettuare un ordine dei componenti mancanti in relazione all'acquisto non concluso. In ogni caso l'acquisto viene registrato perchè il cliente paga comunque al momento dell'ordine (se l'ordine non dovesse andare a buon fine il cliente avrà diritto ad un rimborso) .

FRASI RELATIVE AD ORDINE

Un **ordine** è identificato dall'ID dell'acquisto non concluso, da una data in cui l'ordine è stato fatto e da una di arrivo previsto dell'ordine.

FRASI RELATIVE A MANUTENZIONE

Infine il negozio ha la necessità di raccogliere e mantenere le informazioni necessarie alla **manutenzione** dei componenti, e ne riporta quindi la data della avvenuta manutenzione e il tipo di manutenzione (riparazione o assemblaggio di componenti). Nella manutenzione viene tenuto conto poi della durata di tale manutenzione, al fine di retribuire i lavoratori con un supplemento a ore oltre allo stipendio fisso mensile.

3 Progettazione concettuale

3.1 Descrizione entità e relazioni

3.1.1 Entità ed attributi ¹

Negozio:

- ◆ ID : string
- ◇ Indirizzo : string
- ◇ Città : string

Magazzino:

- ◆ ID : string
- ◇ Indirizzo : string
- ◇ Città : string
- ◇ Volume : int

Componente:

- ◆ ID : string
- ◇ Nome : string
- ◇ Prezzo : float
- ◇ Tipo : enum{SM,CPU,SV,Cooler,RAM,Mem,Case,Alim}

Dove SM = scheda madre, SV = scheda video, Mem = memoria, Alim = alimentatore.

Produttore:

- ◆ Nome : string
- ◇ Email : string
- ◇ Telefono : string

PC:

- ◆ Nome : string
- ◇ Prezzo : float
- ◇ Funzione : enum{Ufficio, Gaming, Grafica e design}

Dipendenti:

- ◆ CF : string
- ◇ Nome : string
- ◇ Cognome : string
- ◇ Data : date
- ◇ Stipendio : int

I dipendenti si suddividono in Commessi, Manutentori e Consulenti, ognuno dei quali non ha bisogno di informazioni aggiuntive per essere rappresentato.

Cliente:

¹Gli identificatori delle entità sono indicati dal simbolo ◆ mentre gli altri attributi sono indicati dal simbolo ◇

- ◆ ID : string
- ◇ Nome : string
- ◇ Cognome : string
- ◇ Email : string

Acquisto:

- ◆ ID : string
- ◇ Totale : float
- ◇ Data : date

Gli acquisti a loro volta si suddividono in acquisti che riguardano componenti o che riguardano pc già assemblati. Gli acquisti di componenti poi si suddividono in acquisti conclusi e non conclusi. Ciò per rappresentare il fatto che se alcuni componenti dovessero mancare in magazzino, il cliente può procedere con l'ordinazione di quest'ultimi.

Ordine:

- ◆ ID : int
- ◇ Date_ordine : date
- ◇ Data_arrivo : date

Manutenzione:²

- ◆ ID_manutentore : string
- ◆ Data : timestamp
- ◇ Tipo : enum{Riparazione, Assemblaggio}
- ◇ Durata : time

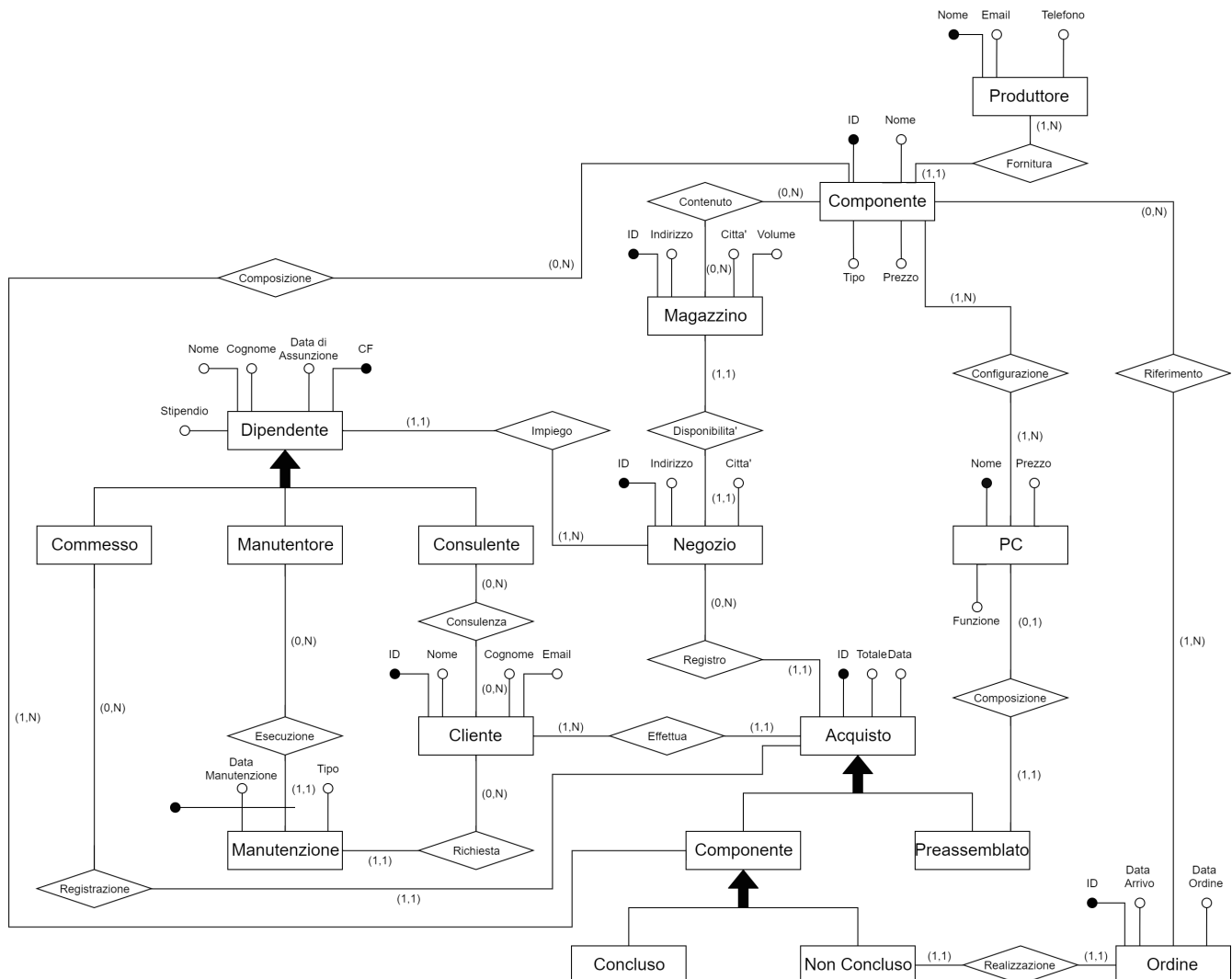
3.1.2 Relazioni

- Negozio - Magazzino : **disponibilità**
 - Ogni negozio dispone di un singolo magazzino
 - Un magazzino rifornisce un singolo negozio
- Negozio - Dipendente : **impiego**
 - Ciascun negozio ha almeno un dipendente
 - Un dipendente lavora in un singolo negozio
 - Il negozio può avere più di un dipendente
- Negozio - Acquisto : **registro**
 - Il negozio può non aver registrato alcun acquisto o averne registrati molteplici
 - Ogni acquisto viene identificato da un id univoco tra i negozi, perciò può essere fatto da un negozio solo
- Magazzino - Componente : **contenuto**
 - Un magazzino contiene da 0 a molti componenti
 - Un componente può essere contenuto in più magazzini
- Componente - Produttore : **fornitura**
 - Ciascun componente è prodotto e fornito da un produttore
 - Un produttore produce e fornisce almeno un componente
- Componente - PC : **configurazione**
 - Più componenti formano la configurazione di un pc
 - Un componente può essere inserito in più di un pc
- Consulente - Cliente : **consulenza**
 - Un consulente può non aver effettuato consulenze o averne effettuate molte
 - Un cliente può aver ricevuto nessuna consulenza oppure diverse
- Manutentore - Manutenzione : **esecuzione**

²L'entità ha un identificatore esterno con l'entità manutentore

- ☐ Un manutentore ha la possibilita' di eseguire piu' manutenzioni o anche nessuna
- ☐ Una manutenzione deve essere eseguita da un manutentore
- Manutenzione - Cliente : **richiesta**
 - ☐ Una manutenzione e' richiesta da un cliente
 - ☐ Un cliente puo' richiedere da 0 a molte manutenzioni
- Acquisto - Cliente : **effettua**
 - ☐ Ciascun acquisto e' realizzato da un cliente
 - ☐ Un cliente effettua almeno un acquisto e puo' farne anche molteplici
- Acquisto - Commesso : **registrazione**
 - ☐ Un commesso registra da 0 a molti acquisti
 - ☐ Un acquisto e' registrato da un cliente
- Acquisto(Componente) - Componente : **composizione**
 - ☐ Un acquisto di componenti e' composto da almeno un componente
 - ☐ Un componente puo' essere compreso in piu' di un acquisto o in nessuno
- Acquisto(Componente) Non concluso - Ordine : **realizzazione**
 - ☐ Un acquisto non concluso di un componente implica la realizzazione di un ordine
 - ☐ Un ordine e' realizzato a fronte di un singolo acquisto non concluso
- Acquisto(Pre-assemblato) - PC : **composizione**
 - ☐ Un acquisto di un pc pre-assemblato e' composto da un singolo PC
 - ☐ Un PC puo' essere registrato o meno in un acquisto(pre-assemblato)
- Ordine - Componente : **riferimento**
 - ☐ Un ordine riferisce a uno o piu' componenti
 - ☐ Un componente puo' essere riferito in piu' di un ordine o da nessuno

3.2 Schema concettuale (E-R)



4 Progettazione logica

4.1 Ristrutturazione dello schema

4.1.1 Analisi delle ridondanze

1. Inserimento di un attributo **Totale** in **Acquisto**:

Si tratta di un attributo derivato calcolabile dalla somma dei singoli prezzi dei **componenti** che formano l'acquisto.

Si considerano le seguenti due operazioni che coinvolgono la base di dati, per analizzare la sua efficienza in presenza o meno dell'attributo ridondante:

- Operazione 1) Registrazione di un nuovo acquisto nella base di dati.

- Operazione 2) Lettura giornaliera del fatturato, ossia del totale del denaro riscattato in seguito alle vendite giornaliere.

Si suppone che in media vengano registrati **10 acquisti di componenti**, ciascuno dei quali sia formato da un numero significativo di componenti, adatto ad assemblare correttamente un pc. Verranno considerati quindi **8 componenti** per ogni acquisto.

- In caso di **Presenza dell'attributo ridondante**:

	Concetto	Costrutto	Accessi	Tipo
Operazione 1)	Acquisto	Entita'	1	S
	Acquisto_Componenti	Entita'	1	S
	Composizione	Relazione	8	S
	Componente	Entita'	8	L

Si attribuisce alle operazioni in scrittura un peso doppio rispetto a quelle in lettura, quindi un singolo acquisto per essere registrato porta ad un costo complessivo di $(1 * 2) + (1 * 2) + (8 * 2) + 8 = 28$ **accessi**. Questa operazione abbiamo definito essere eseguita in media circa 10 volte al giorno, quindi porta ad un totale di **280 accessi** giornalieri.

	Concetto	Costrutto	Accessi	Tipo
Operazione 2)	Acquisto_Componenti	Entita'	1	L
	Acquisto	Entita'	1	L

Il costo totale dell'operazione 2 e' dunque di $(1 + 1) * 10 = 20$ **accessi**.

- In caso di **Assenza dell' attributo ridondante**:

	Concetto	Costrutto	Accessi	Tipo
Operazione 1)	Acquisto	Entita'	1	S
	Acquisto_Componenti	Entita'	1	S
	Composizione	Relazione	8	S
	Componente	Entita'	8	L

Si noti che in assenza dell'attributo ridondante, l'operazione 1 richiede esattamente gli stessi accessi che tenevano conto dell'attributo. Questo perche' l'inserimento di un acquisto e' indipendente da tale attributo.

	Concetto	Costrutto	Accessi	Tipo
Operazione 2)	Acquisto_componenti	Entita'	1	L
	Composizione	Relazione	8	L
	Componente	Entita'	8	L

Si denota dunque che con un totale di $(1 + 8 + 8) * 10 = 170$ **accessi**, l'assenza di un attributo ridondante per il totale del prezzo della vendita porta a dei rallentamenti nell'interrogazione del database in caso di operazioni attuate quotidianamente.

In seguito a tale analisi si decide di mantenere la presenza dell'attributo ridondante **Totale** nell'entita' Acquisto.

2. Inserimento di un attributo **Prezzo** in PC:

Si tratta di un attributo derivato calcolabile dalla somma dei singoli prezzi dei **componenti** che compongono la configurazione del PC e il costo relativo all'assemblaggio dei componenti. Quest'ultimo si riferisce al costo aggiuntivo che il negozio deve pagare al momento dell'acquisto di un PC pre-assemblato. In genere tale costo aggiuntivo risulta essere il 25% del prezzo totale dei componenti presenti nel pc.

Si considerino dunque le seguenti operazioni:

- Operazione 1) Registrazione nel database di un nuovo PC pre-assemblato.
- Operazione 2) Calcolo del fatturato delle vendite relative ai PC pre-assemblati.

Anche in questo caso si suppone che i componenti che formano un pc siano in media 8. La frequenza media stimata per l'esecuzione della prima operazione e' di 3 volte al giorno. Si presume dunque che vengano registrate tante vendite quante sono le registrazioni di nuovi PC pre-assemblati.

- In caso di **Presenza dell'attributo ridondante:**

	Concetto	Costrutto	Accessi	Tipo
Operazione 1)	PC	Entita'	1	S
	Configurazione	Relazione	8	S
	Componente	Entita'	8	L

Dunque ogni registrazione di un pc porta ad un numero pari a $(1 * 2) + (8 * 2) + 8 = 26$ **accessi**, che considerando la frequenza con cui tale operazione viene eseguita, porta ad un totale di $26 * 5 = 130$ **accessi** giornalieri.

	Concetto	Costrutto	Accessi	Tipo
Operazione 2)	Acquisto_PC	Entita'	1	L
	Composizione	Relazione	1	L
	PC	Entita'	1	L

Quest'operazione porta dunque ad un costo di $(1 + 1 + 1) * 5 = 15$ **accessi** giornalieri.

- In caso di **Assenza dell' attributo ridondante:**

Per l'operazione 1) analogamente a quanto visto in precedenza nell'analisi di ridondanza per l'inserimento di un attributo totale ad acquisto, il numero di accessi non dipende dalla presenza dell'attributo ridondante e non aggiunge quindi complessita' all'operazione. Il numero di accessi risulta dunque essere anche in questo caso pari a **130 accessi**.

	Concetto	Costrutto	Accessi	Tipo
Operazione 2)	Acquisto_PC	Entita'	1	L
	Composizione	Relazione	1	L
	PC	Entita'	1	L
	Configurazione	Relazione	8	L
	Componente	Entita'	8	L

Si denota dunque come, anche in questo caso, in assenza di un attributo ridondante il numero di accessi giornalieri sia di gran lunga maggiore rispetto al caso in cui vi sia l'attributo ridondante. Per l'operazione infatti sono necessari $(1 + 1 + 1 + 8 + 8) * 5 = 95$ **accessi** rispetto ai 15 calcolati in precedenza. Si manterra' dunque la presenza di un attributo **Prezzo** nell'entita' PC.

Questo campo inoltre, per rappresentare il totale di un acquisto, corrispondera' al campo totale nell'entita' acquisti, dove l'acquisto si riferisce a PC pre-assemblati.

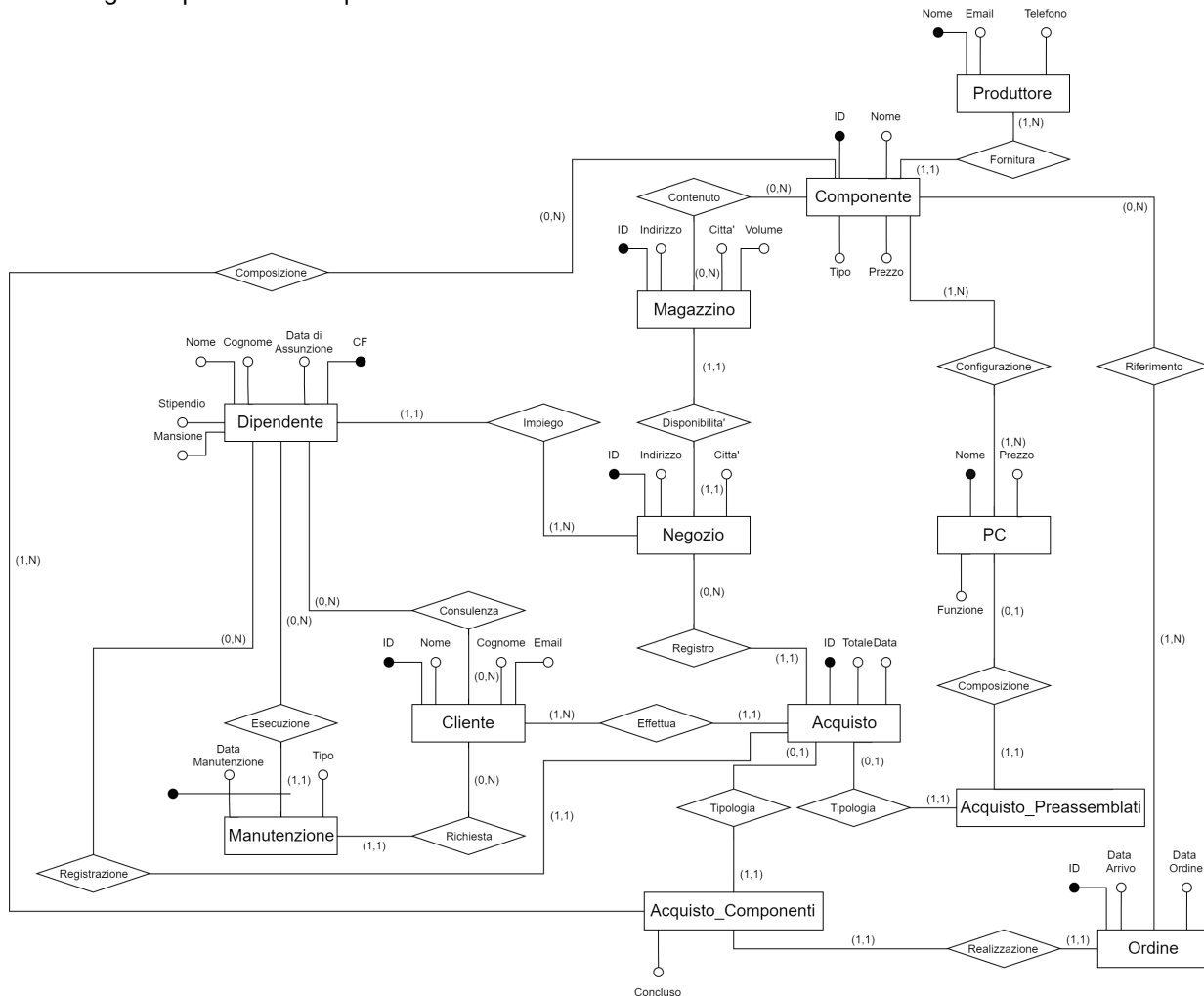
4.1.2 Politiche di ristrutturazione dello schema

Lo schema, a seguito dell' analisi delle ridondanze e di alcune considerazioni verra' ristrutturato tenendo conto dei seguenti punti fondamentali:

- Aggiunta di attributi ridondanti a seguito delle analisi riportate nella precedente sezione.
- Eliminazione della generalizzazione riguardante l' entita' padre Dipendente attraverso l'**accorpamento** delle entita' figlie al padre e l' aggiunta di un relativo campo '**Mansione**'.

- Eliminazione della generalizzazione riguardante l' entità Acquisto, mediante la **creazione di relazioni** tra Acquisto e le sue entità figlie Acquisto_Componenti e Acquisto_Preassemblati.
- Eliminazione della sottogeneralizzazione di Acquisto_Componenti mediante l' **accorpamento** delle entità figlie Concluso e Non concluso al padre. Viene aggiunto un campo Concluso di tipo bool per rappresentare la differenza tra i tipi di Acquisto_Componenti.

In seguito riportiamo dunque lo schema ristrutturato:



5 Schema Relazionale

5.1 Tabelle

NEGOZIO (ID, Indirizzo, Citta')

MAGAZZINO (ID, Indirizzo, Citta', Volume, Negozio)

CONTENUTO_MAGAZZINO (Magazzino, Componente)

DIPENDENTE (CF, Nome, Cognome, Stipendio, Data.Assunzione, Negozio)

MANUTENZIONE (Dipendente, Data, Cliente, Tipo)

CONSULENZA (Dipendente, Cliente)

CLIENTE(ID, Nome, Cognome, Email)
ACQUISTO (ID, Data, Totale, Negozio, Dipendente, Cliente)
ACQUISTO_COMPONENTI (ID_acquisto, Concluso)
COMPOSIZIONE_ACQUISTO (ID_acquisto, Componente)
ACQUISTO_PREASSEMBLATI (ID_acquisto, NomePC)
ORDINE (ID, Data_Ord, Data_Arr, Acquisto.componente)
PC (Nome, Prezzo, Funzione)
CONFIGURAZIONE (NomePC, Componente)
COMPONENTE (ID, Nome, Tipo, Prezzo, Produttore)
REGISTRO_ORDINI (Ordine, Componente)
PRODUTTORE (Nome, Email, Telefono)

5.2 Chiavi esterne

Magazzino.Nwgozio ——> Negozio.ID
Contenuto_Magazzino.Magazzino ——> Magazzino.ID
Contenuto_Magazzino.Componente ——> Componente.ID
Dipendente.Negozio ——> Negozio.ID
Manutenzione.Cliente ——> Cliente.ID
Manutenzione.Dipendente ——> Dipendente.CF
Consulenza.Dipendente ——> Dipendente.CF
Consulenza.Cliente ——> Cliente.ID
Acquisto.Negozio ——> Negozio.ID
Acquisto.Dipendente ——> Dipendente.CF
Acquisto.Cliente ——> Cliente.ID
Acquisto_Componenti.ID_Acquisto ——> Acquisto.ID
Composizione_Acquisto.ID_Acquisto ——> Acquisto.ID
Composizione_Acquisto.Componente ——> Componente.ID
Acquisto_Preassemblati.ID_Acquisto ——> Acquisto.ID
Acquisto_Preassemblati.NomePC ——> PC.Nome
Ordine.Acquisto_Componente ——> Acquisto_Componenti.ID_Acquisto
Configurazione.NomePC ——> PC.Nome
Configurazione.Componente ——> Componente.ID
Componente.Produttore ——> Produttore.Nome
Registro_Ordini.Ordine ——> Ordine.ID
Registro_Ordini.Componente ——> Componente.ID

6 Query e Indici

6.1 Query

Query 1) Restituisce Nome,Cognome,CF di manutentori che hanno eseguito almeno una manutenzione nel mese corrente. Viene inoltre mostrata la durata in un formato ore_minuti di tale manutenzione affinche' il manutentore possa essere retribuito in base al tempo dedicato a tali manutenzioni.

```
SELECT D.Nome, D.Cognome, D.CF, sum(EXTRACT(HOURL,MINUTE FROM M.Durata)) as Durata  
FROM Manutenzione as M JOIN Dipendente as D ON M.Dipendente = D.CF
```

```
WHERE EXTRACT(YEAR FROM M. Data) = EXTRACT(YEAR FROM CURDATE())
      AND EXTRACT(MONTH FROM M. Data) = EXTRACT(MONTH FROM CURDATE())
GROUP BY D.Nome;
```

Nome	Cognome	CF	Durata
Federico	Salvi	SLVFDR98C21G867D	234
Guido	Tirano	TRNGUD93E25Q829R	37

Viene dunque mostrato che Federico Salvi ha eseguito una o piu' manutenzioni nell'arco del mese corrente per una durata di 2:34 ore. Analogamente viene mostrato quanto tempo ha lavorato ad una o piu' manutenzioni Guido Tirano.

Query 2) Restituisce Nome, Cognome e Email di tutti i clienti che non hanno mai effettuato un acquisto o che non lo hanno effettuato in particolare negli ultimi 6 mesi. Questo permette poi al negozio di inviare delle email promozionali a tali clienti per invogliarli a procedere con un acquisto.

```
SELECT C.Nome, C.Cognome, C.Email
FROM Cliente as C LEFT JOIN Acquisto as A ON C.ID = A.Cliente
WHERE C.ID NOT IN (SELECT C.ID
                   FROM Cliente as C JOIN Acquisto as A ON C.ID = A.Cliente
                   WHERE EXTRACT(YEAR FROM A.Data) =
                        EXTRACT(YEAR FROM DATE.SUB(CURDATE(), INTERVAL 6 MONTH))
                   GROUP BY C.ID
                  );
```

Nome	Cognome	Email
Barbara	Troni	BT@gmail.com
Carlo	Rossi	CR@gmail.com
Fabio	Fumelli	FF@gmail.com
Harry	Gentilon	HG@gmail.com
Norberto	Bola	NB@gmail.com
Simona	Certosini	SC@gmail.com
Tancredi	Yong	TY@gmail.com
Zama	Ermes	ZE@gmail.com

Con la query annidata si vanno a ricercare tutti i clienti che **hanno** effettuato un acquisto negli ultimi 6 mesi (estraendo le date degli acquisti e confrontandole con quella corrente), successivamente si vanno a scegliere tutti gli ID di clienti che **non** sono presenti in tale selezione.

Query 3) Per ogni negozio restituisce ID, Nome e Produttore dei componenti venduti piu' di una volta da tale negozio.

```
SELECT A.Negozio, C.ID, C.Nome, count(*) as Vendite
FROM Acquisto as A JOIN Acquisto_Componenti as AC ON A.ID = AC.ID_Acquisto
      JOIN Composizione_Acquisto as CA ON AC.ID_Acquisto = CA.ID_Acquisto
      JOIN Componente as C ON CA.Componente = C.ID
GROUP BY C.ID
HAVING Vendite > 1
ORDER BY Negozio;
```

Negozio ▲ 1	ID	Nome	Vendite
PD024	N1060	GeForce GTX 1060	2
PD024	A3600	AMD Ryzen 5 3600	2
PD089	AS450	STRIX B450-F	2
PD089	CH100	H100i PRO	2
PD089	CVLPX	Vengeance LPX 16GB	2
VI055	MXSOC	MSI Ventus XS OC	2
VI055	Q300L	MasterBox Q300L	2

Query 4) Ritorna ID, Indirizzo dei negozi, insieme alla media degli stipendi recepiti dai dipendenti e al totale degli acquisti registrati per ogni negozio appartenente alla catena.

```

DROP VIEW IF EXISTS TotAcquisti;
CREATE VIEW TotAcquisti(Negozio, NumeroAcquisti) as
SELECT Negozio, count(*)
FROM Acquisto
GROUP BY Negozio;

SELECT N.ID, N.Indirizzo, avg(Stipendio) as Media_Stipendi,
       TA.NumeroAcquisti as TotAcquisti
FROM Dipendente as D JOIN Negozio as N ON D.Negozio = N.ID
     JOIN Acquisto as A ON N.ID = A.Negozio
     JOIN TotAcquisti as TA ON TA.Negozio = N.ID
GROUP BY N.ID;

```

ID	Indirizzo	Media_Stipendi	TotAcquisti
AB126	Via Principi 126	1700.0000	1
AL041	Via Savona 41	1700.0000	1
PD024	Via Altinate 24	1283.3333	3
PD089	Via Gattamelata 89	1633.3333	1
VI055	Via Greco 55	1633.3333	1

La query è realizzata mediante l'utilizzo di una vista di supporto che seleziona il numero totale di acquisti registrati da ciascun negozio. La query successivamente utilizza questa vista per restituire il totale degli acquisti, oltre alla media degli stipendi per ogni esercizio commerciale.

Query 5) Restituisce i componenti relativi all'acquisto '17_0000003' che non sono presenti in magazzino e che dovranno dunque essere ordinati.

```

SELECT C.ID, C.Nome
FROM Composizione_Acquisto as CA JOIN Componente as C ON CA.Componente = C.ID
WHERE CA.ID_Acquisto = '17_0000003' AND
      C.ID NOT IN (SELECT CM.Componente
                  FROM Acquisto as A JOIN Magazzino as M ON A.Negozio = M.Negozio
                  JOIN Contenuto_Magazzino as CM ON CM.Magazzino = M.ID
                  WHERE A.ID = '17_0000003'
                 );

```

ID	Nome
AS450	STRIX B450-F
CH100	H100i PRO

La sotto-query prende in esame tutti i componenti contenuti nel magazzino corrispondente al negozio dove e' stato registrato l'acquisto. Successivamente si vanno a selezionare gli ID dei componenti che formano l'acquisto, che pero' non sono riportati dalla sotto-query e che quindi non sono presenti in magazzino.

Query 6) Restituisce ID, Nome e Prezzo di tutti i componenti registrati che risultano avere un prezzo minore della media del totale degli acquisti di componenti formati da un singolo articolo.

```
DROP VIEW IF EXISTS AcquistiSingoli ;
```

```
CREATE VIEW AcquistiSingoli (ID_Acquisto , ID_Componente , Totale) as  
SELECT AC.ID_Acquisto , CA.Componente , A.Totale  
FROM Acquisto_Componenti as AC JOIN Composizione_Acquisto as CA  
    ON AC.ID_Acquisto=CA.ID_Acquisto  
    JOIN Acquisto as A ON AC.ID_Acquisto = A.ID  
GROUP BY CA.ID_Acquisto  
HAVING count(CA.Componente) = 1 ;
```

```
SELECT C.ID , C.Nome , C.Prezzo  
FROM Componente as C  
WHERE C.Prezzo < (SELECT AVG( AcquistiSingoli.Totale )  
                FROM AcquistiSingoli  
                );
```

Anche questa query (analogamente a quanto succede nella numero 4) si serve di una vista. Tale vista seleziona gli acquisti formati da un singolo componente e ne riporta anche il totale. Questo per permettere alla query di confrontare il prezzo di tutti i componenti registrati con la media dei totali riportati dalla vista(quindi di ogni acquisto singolo).

L'output della query per mancanza di spazio viene riportato interamente nella pagina seguente.

ID	Nome	Prezzo
A3600	AMD Ryzen 5 3600	194.99
AS450	STRIX B450-F	117.99
C200R	Corsair 200R	109.99
CH100	H100i PRO	73.99
CM212	Hyper 212 EVO	34.49
CRM19	Corsair RM2019	109.99
CSCXM	Corsair CXM	73.98
CVLPX	Vengeance LPX 16GB	74.67
I660P	660P Series	109.99
I9700	Intel Core i7-9700	369.99
MB450	B450 TOMAHAWK	114.99
MXSOC	MSI Ventus XS OC	297.99
Q300L	MasterBox Q300L	44.99
S860E	860 EVO	77

6.2 Indici

Ipotizzando che il database di **Hardware House** sia di dimensioni molto elevate e in particolare:

- La tabella **Componente** contenga al suo interno molte tuple
- I componenti nuovi introdotti nel mercato siano in misura non troppo elevata
- La ricerca dei componenti e' un' operazione svolta con una frequenza sufficientemente alta

Si considera l'indicizzazione della tabella **Componente** al fine di velocizzare le operazioni di ricerca che coinvolgono quest'ultimi.

In particolare la maggior parte delle ricerche interessa maggiormente la tipologia del componente, piu' che il suo identificativo o il suo nome, dunque l'indicizzazione averra' non su l'intera tabella ma sull' attributo **Tipo**.

Il codice che interessa la creazione degli indici e' il seguente:

```
CREATE INDEX idx_TipoComponenti ON Componente( Tipo );
```