

## **Progettazione e implementazione di una base di dati per la gestione dell'ufficio acquisti di un ente pubblico**

DARIO SIMONE 153505 dario.simone@spes.uniud.it

DITTARO FEDERICO 154538 dittaro.federico@spes.uniud.it

MAZZEGA GABRIELE 152937 mazzega.gabriele@spes.uniud.it

MONTE STEFANO 152122 monte.stefano@spes.uniud.it

*Progetto Basi di Dati anno accademico 2022/2023*

*Corso di laurea in Informatica, Università degli studi di Udine*

<b>1.1 Studio della richiesta</b>	<b>3</b>
1.1.1 Richiesta originale	3
1.1.2 Riscrittura e strutturazione dei requisiti	4
<b>1.2 Requisiti strutturati</b>	<b>5</b>
1.2.1 Glossario	5
<b>1.3 Requisiti operazionali (Operazioni frequenti all'interno della base di dati)</b>	<b>5</b>
<b>2.1 Costruzione dello schema Entità - Relazioni</b>	<b>6</b>
2.1.1 Responsabile e Dipartimento	6
2.1.2 Dipartimento e Richiesta	7
2.1.3 Richiesta e articolo	7
2.1.4 Articolo e Fornitore	8
2.1.5 Richiesta e Ordine	9
2.1.6 Fornitore e Ordine	10
<b>2.2 Schema generale</b>	<b>11</b>
<b>3.1 Tabella dei volumi</b>	<b>12</b>
<b>3.2 Analisi delle ridondanze</b>	<b>13</b>
3.2.1 Tabella degli accessi in assenza di ridondanza	14
3.2.2 Tabella degli accessi in presenza di ridondanza	14
3.2.3 Considerazioni sulla ridondanza	15
<b>3.3 Analisi dei cicli</b>	<b>15</b>
<b>3.4 Rimozione dei costrutti</b>	<b>16</b>
3.4.1 Generalizzazioni	16
3.4.2 Attributi multivalore	16
3.4.3 Partizionamenti	16
3.4.4 Relazioni n-arie	16
3.4.5 Scelta delle chiavi primarie	17
<b>3.5 Schema E-R ristrutturato</b>	<b>17</b>
<b>3.6 Schema Logico</b>	<b>17</b>
3.6.1 Modello relazionale	17
3.6.2 Vincoli	18
<b>4.1 Definizioni delle relazioni in SQL</b>	<b>20</b>
<b>4.2 Analisi e scelta degli indici</b>	<b>24</b>
<b>5.1 Inserimento dei dati</b>	<b>25</b>
<b>5.2 Definizione dei Trigger</b>	<b>29</b>
5.2.1 Prezzo articolo	30
5.2.2 Cancellazione responsabile	30
<b>5.3 Definizione di Query</b>	<b>31</b>
<b>6.1 Numero di fornitori che forniscono un dato articolo</b>	<b>33</b>
<b>6.2 Totale articoli richiesti per dipartimento</b>	<b>34</b>
<b>6.3 Prezzo totale delle richieste per dipartimento</b>	<b>35</b>

# 1 Raccolta e analisi dei requisiti

## 1.1 Studio della richiesta

### 1.1.1 Richiesta originale

Si vuole realizzare una base di dati per la gestione dell'ufficio acquisti di un ente pubblico caratterizzato dal seguente insieme di requisiti:

- l'ente sia organizzato in un certo insieme di dipartimenti, ciascuno identificato univocamente da un codice e caratterizzato da una breve descrizione e dal nominativo del responsabile (si assuma che ogni dipartimento abbia un unico responsabile e che una stessa persona possa essere responsabile di più dipartimenti)
- ogni dipartimento possa formulare delle richieste d'acquisto; ogni richiesta di acquisto formulata da un dipartimento sia caratterizzata da un numero progressivo, che la identifica univocamente all'interno dell'insieme delle richieste del dipartimento (esempio, richiesta numero 32 formulata dal dipartimento D37), da una data (si assuma che uno stesso dipartimento possa effettuare più richieste in una stessa data), dall'insieme degli articoli da ordinare, con l'indicazione, per ciascun articolo, della quantità richiesta, e dalla data prevista di consegna;
- ogni articolo sia identificato univocamente da un codice articolo e sia caratterizzato da una breve descrizione, da una unità di misura e da una classe merceologica;
- ogni fornitore sia identificato univocamente da un codice fornitore e sia caratterizzato dalla partita IVA, dall'indirizzo, da uno o più recapiti telefonici e da un indirizzo di posta elettronica; alcuni fornitori (non necessariamente tutti) possiedano un numero di fax;
- ad ogni fornitore sia associato un listino, comprendente uno o più articoli; per ciascun articolo appartenente ad un dato listino siano specificati il codice articolo, il prezzo unitario, il quantitativo minimo d'ordine e lo sconto applicato;
- per soddisfare le richieste provenienti dai vari dipartimenti, l'ufficio acquisti emetta degli ordini; ogni ordine sia identificato univocamente da un codice ordine e sia caratterizzato dalla data di emissione, dal fornitore a cui viene inviato, dall'insieme degli articoli ordinati, con l'indicazione, per ciascuno di essi, della quantità ordinata, e dalla data prevista di consegna (si assuma che un ordine possa fondere insieme più richieste d'acquisto dei dipartimenti).

### 1.1.2 Riscrittura e strutturazione dei requisiti

#### *Frase di natura generale*

Si vuole realizzare una base di dati per la gestione dell'ufficio acquisti di un ente pubblico

#### *Frase relative al dipartimento*

L'ente è organizzato in dipartimenti. Il dipartimento è identificato univocamente da un codice, caratterizzato da una breve descrizione e dal nominativo del responsabile. Ogni dipartimento ha un singolo responsabile.

#### *Frase Relative al responsabile*

Ogni responsabile può essere responsabile di più dipartimenti.

#### *Frase relative alla richiesta d'acquisto*

Ogni dipartimento può formulare delle richieste d'acquisto; ogni richiesta di acquisto è caratterizzata da un numero progressivo, che la identifica univocamente all'interno dell'insieme delle richieste del dipartimento, da una data (un dipartimento può effettuare più richieste in una stessa data), dall'insieme degli articoli da ordinare, con l'indicazione, per ciascun articolo, della quantità richiesta, e dalla data prevista di consegna;

#### *Frase relative all'articolo*

Ogni articolo è identificato univocamente da un codice articolo e caratterizzato da una breve descrizione, da una unità di misura e da una classe merceologica;

#### *Frase relative al fornitore*

Ogni fornitore è identificato univocamente da un codice fornitore e caratterizzato dalla partita IVA, dall'indirizzo, da uno o più recapiti telefonici e da un indirizzo di posta elettronica; alcuni fornitori (non necessariamente tutti) possiedono un numero di fax;

#### *Frase relative al listino*

Ad ogni fornitore sia associato un listino, che comprende uno o più articoli; per ciascun articolo appartenente ad un dato listino è specificati il codice articolo, il prezzo unitario, il quantitativo minimo d'ordine e lo sconto applicato;

#### *Frase relative all'ordine*

Per soddisfare le richieste d'acquisto, l'ufficio acquisti emette degli ordini; ogni ordine è identificato univocamente da un codice ordine ed è caratterizzato dalla data di emissione, dal fornitore a cui viene inviato, dall'insieme degli articoli ordinati, con l'indicazione, per ciascuno di essi, della quantità ordinata, e dalla data prevista di consegna (un ordine può fondere insieme più richieste d'acquisto dei dipartimenti).

## 1.2 Requisiti strutturati

### 1.2.1 Glossario

Termine	Descrizione	Sinonimi	Collegamenti
Dipartimento	diretto da un responsabile, effettua richieste d'acquisto	/	richiesta d'acquisto
Richiesta d'acquisto	un insieme di articoli che un dipartimento vuole ordinare (può essercene più di una per dipartimento)	richiesta	dipartimento articolo ordine
Articolo	articolo contenuto nella richiesta d'acquisto	/	richiesta d'acquisto listino
Fornitore	fornitore di un ordine	/	ordine listino
Ordine	un insieme di richieste d'acquisto fatte ad un fornitore	/	fornitore richiesta d'acquisto
Listino	l'insieme degli articoli venduti da un fornitore	/	fornitore articolo

Tabella 1. Glossario

## 1.3 Requisiti operazionali (Operazioni frequenti all'interno della base di dati)

Di seguito sono descritte le operazioni frequenti all'interno della base di dati: Le prime 2 operazioni sono significative perché utili nell'analisi dei costi di input/output degli accessi alla base di dati in uno scenario realistico; le restanti operazioni sono state individuate per ricoprire sezioni diverse all'interno dello schema concettuale della base di dati.

*Operazione 1:* Interrogare la richiesta d'acquisto per sapere numero di articoli in una richiesta

*Operazione 2:* Aggiungere un articolo ad una richiesta d'acquisto

*Operazione 3:* Cambiare il prezzo unitario di un articolo posseduto da un fornitore

*Operazione 4:* Formulare richieste di acquisto con tutti i suoi dati

*Operazione 5:* Mostrare i dipartimenti controllati da un responsabile

*Operazione 6:* Formulare ordini

*Operazione 7:* Interrogare fornitore sugli articoli che possiede

*Operazione 8:* Valutare prezzo minimo/migliore della richiesta di acquisto

*Operazione 9:* Interrogare la richiesta d'acquisto per sapere numero di articoli in un ordine

*Operazione 10:* Mostrare data di consegna prevista

*Operazione 11:* Riepilogare articoli ordinati

*Operazione 12:* Interrogare l'ordine per sapere il prezzo totale dell'ordine

## 2. Progettazione concettuale

### 2.1 Costruzione dello schema Entità - Relazioni

Per la costruzione dello schema Entità - Relazioni è stata utilizzata la tecnica bottom-up: sono stati realizzati prima dei partizionamenti dello schema E-R per ciascun concetto indipendente e successivamente sono stati collegati fra loro per ottenere lo schema completo del problema iniziale.

Per facilitare l'identificazione delle entità, delle relazioni e per rendere più leggibili i vari diagrammi si è scelto di colorare le entità di azzurro e le relazioni di arancione.

#### 2.1.1 Responsabile e Dipartimento

Il testo specifica la presenza dell'entità DIPARTIMENTO, caratterizzata da un *codice identificativo* che costituisce la chiave primaria e da una breve *descrizione*. Sono stati poi introdotti gli attributi *Nome*, *Ambito* e *Sede* (contenente l'indirizzo del dipartimento). Inoltre, viene specificato che un dipartimento possa essere controllato solo da un responsabile, ma che una stessa persona possa controllare più dipartimenti; quest'ultimo vincolo porta alla creazione dell'entità RESPONSABILE, caratterizzata dal *codice fiscale* che rappresenta la chiave primaria, il *nome*, il *cognome*, il *secondo nome* opzionale, il sesso e la *data di nascita*. Le due entità sono collegate dalla relazione 'Dirige'.

La cardinalità dello schema indipendente è uno-a-molti, in particolare dal lato DIPARTIMENTO è (1,1), in quanto non possono esistere dipartimenti senza responsabili, e allo stesso tempo il numero massimo è uno per i vincoli specificati dal problema, mentre dal lato RESPONSABILE è (1,N) dovuto al fatto che un responsabile può controllare più dipartimenti.

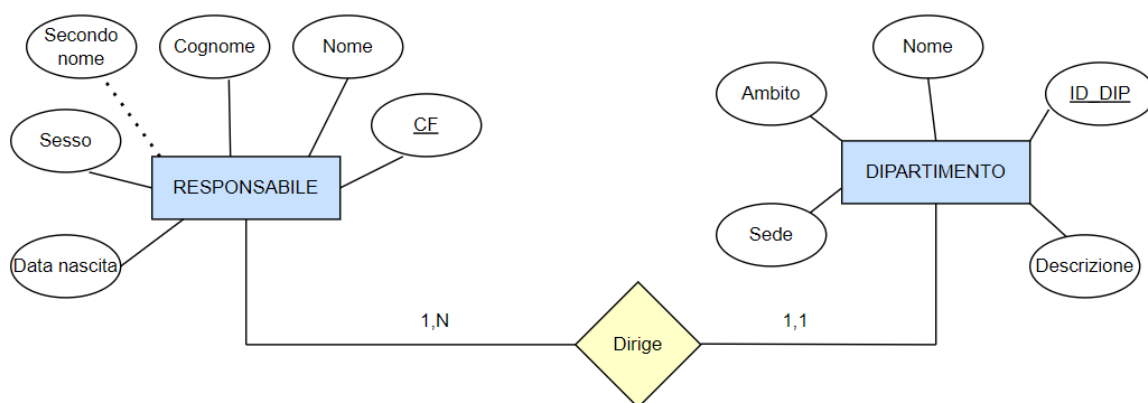


Figura 1: sottoschema della relazione esistente tra responsabile e dipartimento

### 2.1.2 Dipartimento e Richiesta

L'entità RICHIESTA è stata introdotta in quanto il testo specifica che i dipartimenti effettuano delle richieste di acquisto che devono essere distinguibili all'interno del dipartimento stesso. Per quest'ultimo motivo si tratta quindi di un'entità debole rispetto a DIPARTIMENTO (specificata nel paragrafo 2.1.1).

Gli attributi associati all'entità RICHIESTA sono il *numero progressivo*, che rappresenta la chiave e costituisce l'entità debole, e la *data* in cui viene effettuata.

Le due entità sono collegate dalla relazione 'Effettua', una relazione uno-a-molti. Nello specifico, dal lato dell'entità DIPARTIMENTO è (1,N), in quanto un dipartimento può effettuare più richieste, mentre dal lato RICHIESTA è (1,1), dato che la richiesta si riferisce ad uno ed un solo dipartimento.

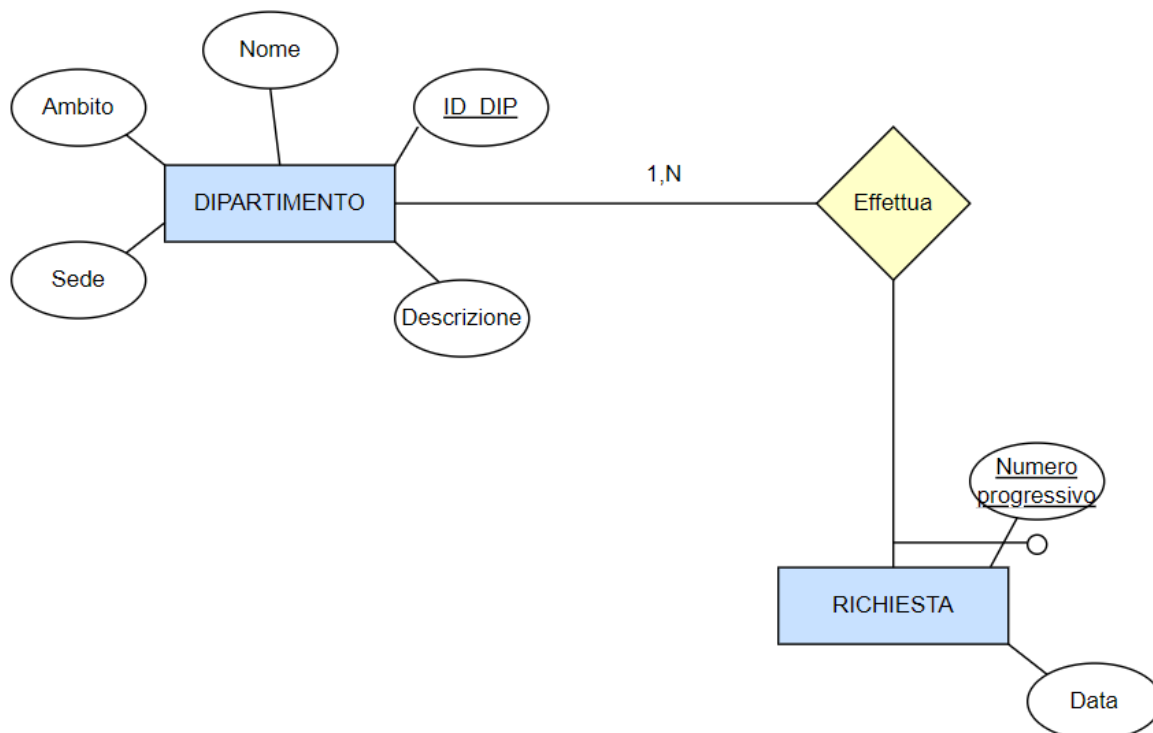


Figura 2: sottoschema della relazione esistente tra dipartimento e richiesta

### 2.1.3 Richiesta e articolo

Una richiesta da parte di un dipartimento deve ovviamente contenere degli articoli e varie informazioni su di essi. L'introduzione dell'entità ARTICOLO e della relazione 'Formata da' permette di gestire queste informazioni, che secondo le specifiche del problema devono includere la *quantità ordinata* per ciascun articolo e la *data prevista di consegna*. Questi due attributi sono collegati alla relazione 'Formata da', permettendo così di gestire questi due vincoli per ogni articolo ordinato.

Gli attributi di ARTICOLO sono *ID\_articolo*, che costituisce la chiave primaria, il *nome*, una breve *descrizione* e la *classe merceologica*.

La relazione introdotta collega le entità ARTICOLO con l'entità RICHIESTA (specificata nel paragrafo 2.1.2) ed è una relazione multi-a-molti con cardinalità (0,N) dal lato di ARTICOLO, poiché può accadere che uno specifico articolo non venga mai richiesto, e cardinalità (1,N) dal lato di RICHIESTA, dato che le richieste contenenti zero articoli sono insensate."

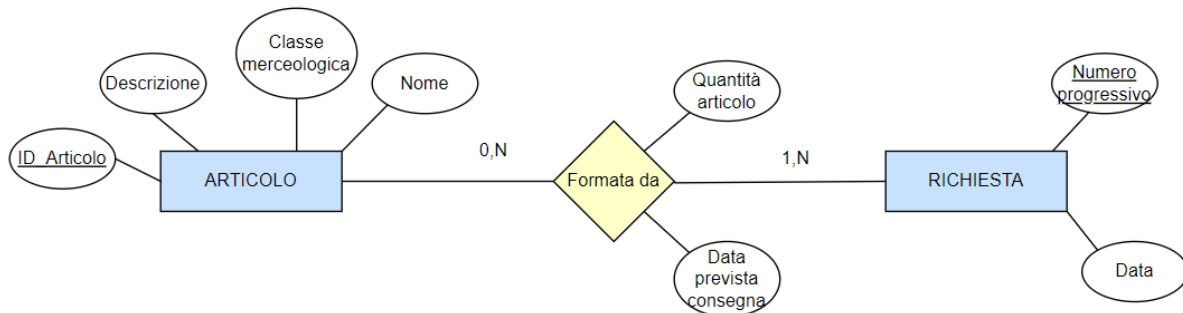


Figura 3: sottoschema della relazione esistente tra dipartimento e richiesta

#### 2.1.4 Articolo e Fornitore

Gli articoli vengono forniti da un FORNITORE, un'altra entità necessaria del diagramma E-R. I suoi attributi sono il *codice identificativo*, che rappresenta la chiave primaria, la *partita IVA*, che data la sua unicità può essere una chiave alternativa, l'*indirizzo* della sede, uno o più *recapiti telefonici* (attributo multivalore), un *indirizzo di posta elettronica* e, opzionalmente, il *numero di fax*.

Inoltre, il testo specifica che ogni fornitore ha associato un listino comprendente uno o più articoli e per ciascuno di essi viene specificato il relativo codice, il *prezzo unitario*, il *quantitativo minimo d'ordine* e lo *sconto applicato*. Per far sì che queste informazioni compaiano nel diagramma, la relazione 'Possiede' che collega il fornitore all'articolo (specificato nel paragrafo 2.1.3) contiene al suo interno questi tre attributi in modo che ogni articolo posseduto dal fornitore contenga queste informazioni.

La relazione appena creata è multi-a-molti e nello specifico ha cardinalità (1,N) sia dal lato FORNITORE, in quanto per essere tale deve possedere uno o più articoli, sia dal lato ARTICOLO, dato che se l'articolo è presente nella base di dati (e quindi è ordinabile da qualche dipartimento) ci deve essere almeno uno o più fornitori che lo forniscono.



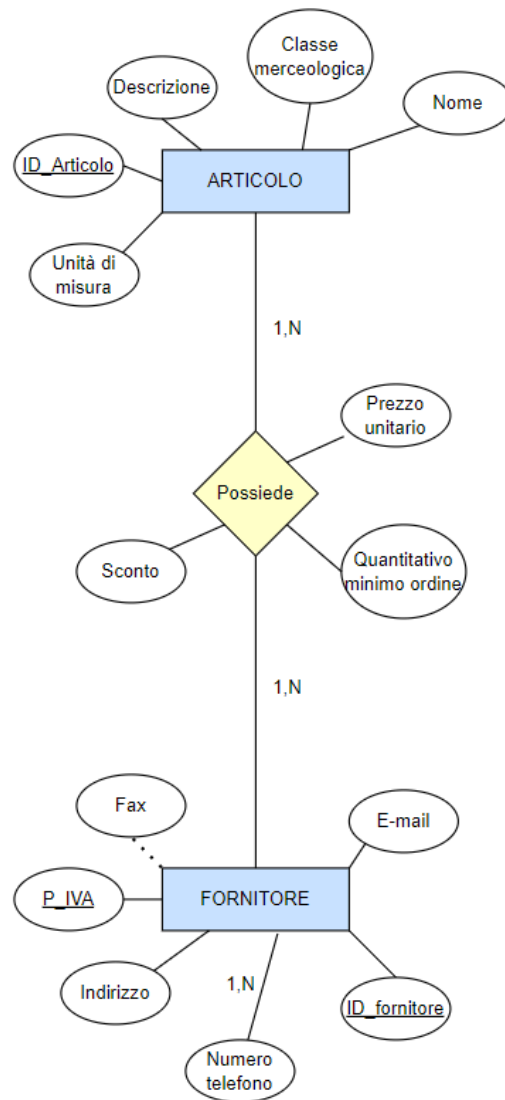


Figura 4: sottoschema della relazione esistente tra articolo e fornitore

### 2.1.5 Richiesta e Ordine

Per soddisfare le varie richieste provenienti dai dipartimenti, l'ufficio acquisti emette degli ordini che, secondo le specifiche del problema, possono fondere insieme più richieste. È pertanto necessario creare questa nuova entità. L'ORDINE è descritto dal *codice\_ordine*, che rappresenta la chiave primaria, e dalla *data di emissione*.

La relazione 'Compone' collega l'entità RICHIESTA (specificata nel paragrafo 2.1.2) all'entità ORDINE. Si tratta di una relazione uno a molti. In particolare, dal lato dell'entità RICHIESTA, la cardinalità è (1,1), in quanto quest'ultima andrà a far parte di uno ed un solo ordine, mentre dal lato di ORDINE è (1,N), dato che ha senso solo se contiene una o più richieste.

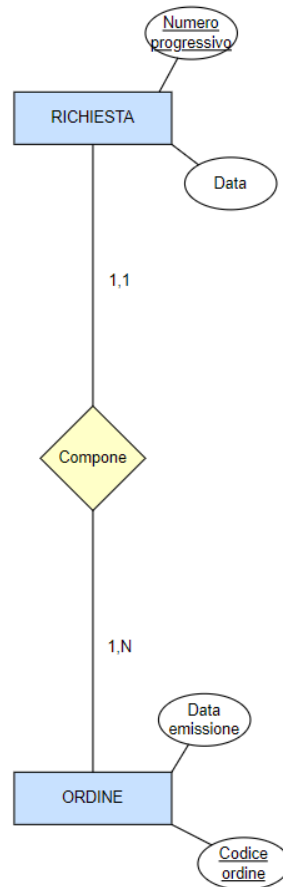


Figura 5: sottoschema della relazione esistente tra richiesta e ordine

### 2.1.6 Fornitore e Ordine

L'ultimo concetto indipendente da modellare riguarda la relazione tra le entità **FORNITORE** (specificato nel paragrafo 2.1.4) e **ORDINE** (specificato nel paragrafo 2.1.5). Un ordine viene inviato ad uno ed un solo fornitore che a sua volta ne può evadere più di uno, si tratta quindi di una relazione uno-a-molti che in questo caso viene chiamata 'Inviato a' e nello specifico ha cardinalità (1,N) dalla parte del **FORNITORE** e (1,1) dalla parte dell'entità **ORDINE**.

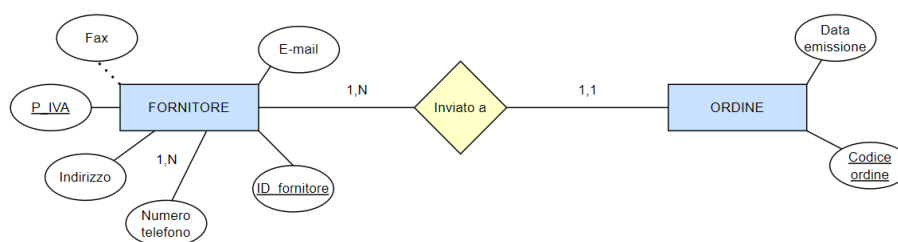


Figura 6: sottoschema della relazione esistente tra richiesta e ordine

## 2.2 Schema generale

Integrando i sottoschemi creati ed esplicitando i vincoli imposti dal problema si ottiene il modello generale della base di dati che può essere utilizzato per sviluppare le fasi successive del progetto.

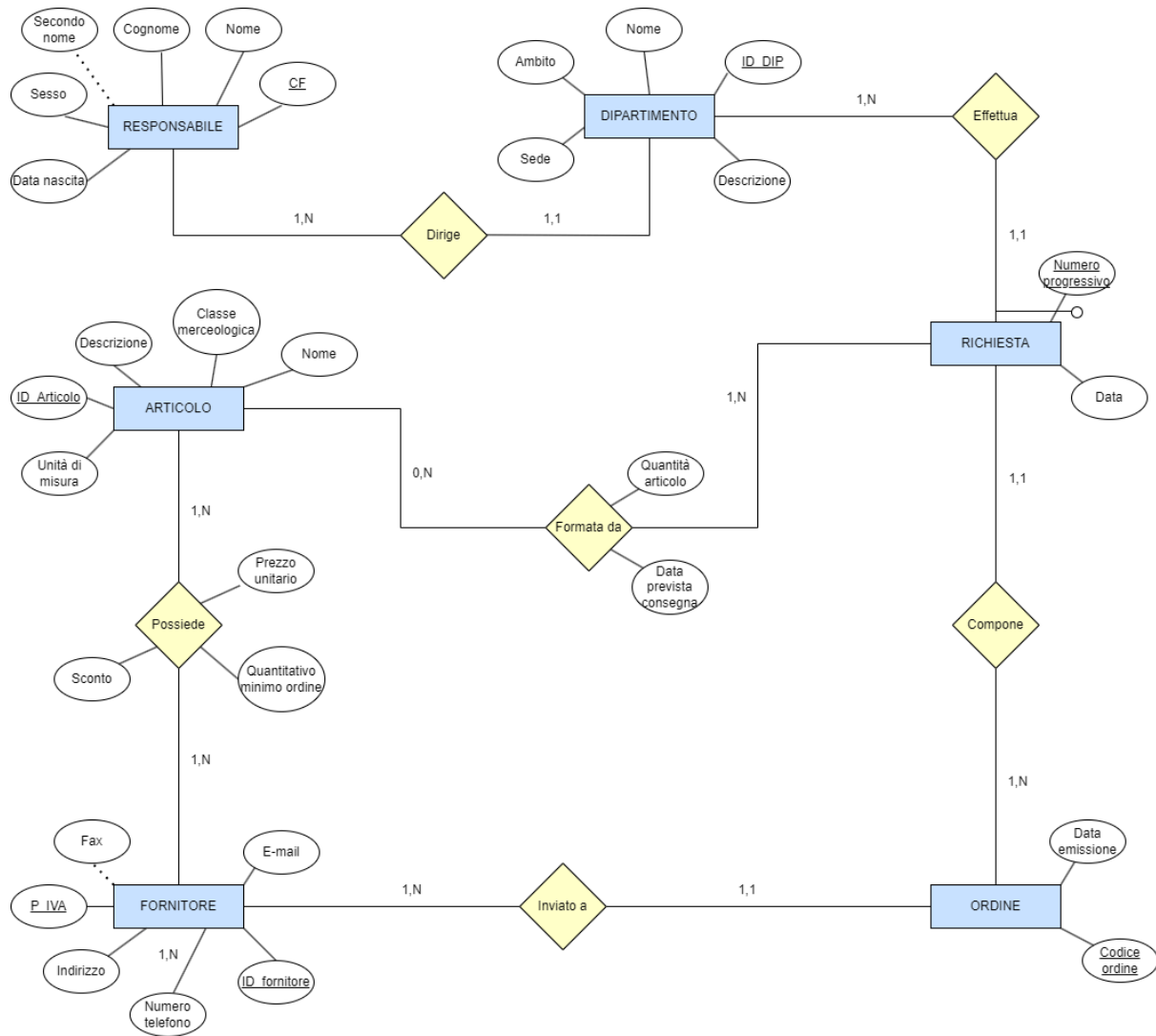


Figura 7: schema concettuale del modello Entità-Relazioni

## 3. Progettazione Logica

### 3.1 Tabella dei volumi

Per condurre un'analisi adeguata delle ridondanze e, di conseguenza, pianificare una ristrutturazione efficace dello schema E-R, è necessario sviluppare una tabella dei volumi. Questa tabella dovrà contenere stime sulla quantità media di ogni oggetto presente nel nostro database. Notiamo che alcuni oggetti mantengono volumi pressoché costanti, mentre altri presentano una crescita graduale. Per gli oggetti soggetti a crescita, considereremo un periodo di riferimento nella nostra base dati di 20 anni, calcolando le quantità previste al punto intermedio del ciclo di vita.

Concetto	Tipo	Volume
RESPONSABILE	Entità	20
Dirige	Relazione	20
DIPARTIMENTO	Entità	20
Effettua	Relazione	10000
RICHIESTA D'ACQUISTO	Entità	10000
Compone	Relazione	10000
ORDINE	Entità	2000
Formata da	Relazione	100000
ARTICOLO	Entità	4000
Inviato a	Relazione	2000
FORNITORE	Entità	150
Possiede	Relazione	15000

Tabella 2: Tabella dei Volumi

La tabella dei volumi (Tabella 2) è stata creata sulla base delle seguenti considerazioni:

L'azienda è suddivisa in 20 DIPARTIMENTI, ciascuno dei quali è guidato da un singolo RESPONSABILE. Di conseguenza, abbiamo un massimo di 20 RESPONSABILI e, quindi, circa 20 coppie <RESPONSABILE, DIPARTIMENTO> nella relazione 'Dirige'.

Ogni dipartimento genera in media 50 richieste d'acquisto all'anno, partendo da zero e tenendo traccia delle richieste d'acquisto passate. A metà del ciclo di vita, ci aspettiamo

circa 10.000 coppie <DIPARTIMENTO, RICHIESTA D'ACQUISTO> nella relazione 'Effettua', corrispondenti a circa 10.000 richieste d'acquisto.

In media, ogni ORDINE è composto da 5 richieste d'acquisto. Pertanto, a metà del ciclo di vita, avremo circa 2.000 ordini, e la cardinalità della relazione 'Compone' sarà di circa 10.000 a metà del ciclo di vita.

Inizialmente l'azienda invia ordini a uno di 100 fornitori e ne aggiunge 5 nuovi ogni anno. Quindi, a metà del ciclo di vita, avremo un totale di 150 fornitori. Poiché ogni ordine è inviato a un unico fornitore, avremo circa 2.000 coppie <ORDINE, FORNITORE> nella relazione 'Inviato a'.

L'azienda aggiunge 200 nuovi articoli alla base di dati ogni anno, partendo da una base iniziale di 2.000 articoli, che si traduce in 4.000 articoli a metà del ciclo di vita. Ogni RICHIESTA D'ACQUISTO, in media, comprende 10 articoli, portando la cardinalità della relazione 'Formata da' a 100.000. Inoltre, ogni fornitore possiede mediamente 100 articoli, il che si traduce in 15.000 coppie nella relazione 'Possiede'.

### 3.2 Analisi delle ridondanze

Si sta attualmente valutando la necessità di aggiungere un attributo derivato *numero articoli* nell'entità RICHIESTA D'ACQUISTO e se tale aggiunta sia effettivamente utile. Gli utilizzi di tale attributo coinvolgono due task principali:

Task 1: Determinare il numero di articoli che compongono una specifica richiesta d'acquisto.

Task 2: Aggiungere o rimuovere un articolo da una richiesta d'acquisto.

Il Task 1 è utile per stimare la quantità complessiva di articoli utilizzati da ciascun dipartimento rispetto agli altri. Di conseguenza, viene eseguito due volte al mese per ogni dipartimento al fine di monitorare con precisione i consumi.

Per quanto riguarda il Task 2, si presume che ogni dipartimento esegua mediamente circa 40 di tali operazioni al mese. Si suppone inoltre che gli articoli inseriti attraverso il Task 2 siano già presenti nella base di dati. Queste informazioni sono sintetizzate nella seguente tabella delle frequenze.

Operazione	Tipo	Frequenza [mese]
Operazione 1	Batch	2
Operazione 2	Interactive	40

Tabella 3: Tabella delle frequenze

### 3.2.1 Tabella degli accessi in assenza di ridondanza

#### Task 1

In assenza di ridondanza, per determinare quanti articoli compongono una specifica richiesta d'acquisto, è necessario scandire la relazione 'Formata da' per individuare tutte le coppie in cui compare la richiesta d'acquisto in questione. Poiché questa relazione contiene in media 100.000 istanze, considerando che questo task si verifica due volte al mese per 20 dipartimenti, il costo annuale totale in termini di accessi in memoria è di:

$$100.000 \cdot 2 \cdot 12 \cdot 20 = 48.000.000 \text{ accessi/anno}$$

Concetto	Tipo	Accessi	Tipologia Accesso
Formata da	Relazione	100000	Read

Tabella 4: Accessi in memoria del Task 1 in assenza di ridondanza

#### Task 2

Per l'aggiunta o la rimozione di un articolo da una richiesta d'acquisto, senza ridondanza, è richiesta un'operazione di scrittura all'interno della relazione 'Formata da'. Questo task si verifica 40 volte al mese per ciascun dipartimento, portando il costo annuale totale in termini di accessi in memoria a:

$$1 \cdot 40 \cdot 12 \cdot 20 = 9.600 \text{ accessi/anno}$$

Concetto	Tipo	Accessi	Tipologia Accessi
Formato da	Relazione	1	Write

Tabella 5: Accessi in memoria del Task 2 in assenza di ridondanza

### 3.2.2 Tabella degli accessi in presenza di ridondanza

#### Task 1

In presenza di ridondanza, per determinare quanti articoli compongono una richiesta d'acquisto, è sufficiente accedere all'attributo *Numero Articoli*, il quale richiede un'operazione di lettura. Il costo complessivo di questo task, che si verifica due volte al mese per ogni dipartimento, ammonta a:

$$1 \cdot 2 \cdot 12 \cdot 20 = 480 \text{ accessi}$$

Concetto	Tipo	Accessi	Tipologia Accessi
RICHIESTA D'ACQUISTO	Entità	1	Read

Tabella 6: Accessi in memoria del Task 1 in presenza di ridondanza

**Task 2**

In presenza di ridondanza, per ogni aggiunta o rimozione di un articolo da una richiesta d'acquisto, è necessario modificare il valore dell'attributo *Numero Articoli*, incrementandolo o decrementandolo di 1. Inoltre, occorre effettuare un'operazione di scrittura all'interno della relazione 'Formata da' per un totale di 2 accessi in memoria. Il costo annuale complessivo di questo task, se si verifica 40 volte al mese per ogni dipartimento, è quindi:

$$2 \cdot 40 \cdot 12 \cdot 20 = 19200 \text{ accessi}$$

Concetto	Tipo	Accessi	Tipologia Accessi
RICHIESTA D'ACQUISTO	Entità	1	Write
Formato da	Relazione	1	Write

Tabella 7: Accessi in memoria del Task 2 in presenza di ridondanza

**3.2.3 Considerazioni sulla ridondanza**

In presenza di ridondanza, l'accesso all'attributo *Numero Articoli* riduce notevolmente il numero di accessi in memoria, come evidenziato dalla differenza significativa tra gli accessi in memoria totali nelle situazioni in cui l'attributo è presente o assente. Questo rende accettabile l'inclusione dell'attributo aggiuntivo *Numero Articoli*, nonostante favorisca task di tipo batch.

Task	Presenza Ridondanza	Assenza Ridondanza
Task 1	48.000.000	480
Task 2	9.600	19.200
Totale	48.009.600	19.680

Tabella 8: Accessi in memoria totali

**3.3 Analisi dei cicli**

La presenza di un ciclo all'interno dello schema richiede una valutazione sulla sua necessità nel contesto dello schema concettuale e se potrebbe rappresentare una ridondanza eliminabile. Per effettuare questa valutazione, si procede rimuovendo una delle relazioni coinvolte e verificando se tale rimozione comporta una perdita d'informazione che non può essere recuperata attraverso la navigazione inversa.

Considerando la relazione 'Compone', se la rimuovessimo, sarebbe possibile risalire a tutti gli ordini in cui un articolo è stato incluso. Tuttavia, non saremmo in grado di determinare quale richiesta specifica ha portato all'inclusione di quell'articolo tra tutti gli ordini in cui è coinvolto.

Analogamente, se eliminassimo la relazione 'Inviato a', potremmo trovarci nella situazione in cui più fornitori sono in grado di soddisfare lo stesso ordine, ma non sapremmo quale di essi ha effettivamente ricevuto l'ordine.

Per quanto riguarda la relazione 'Possiede', se non la considerassimo, sapremmo soltanto quali articoli un fornitore possiede quando questi articoli compaiono negli ordini. Se un articolo non fosse mai incluso in un ordine, il fornitore non ne sarebbe a conoscenza.

Infine, se eliminassimo la relazione 'Formata da', potremmo creare ordini contenenti richieste d'acquisto, ma non avremmo informazioni sugli articoli specifici inclusi in tali ordini.

Queste considerazioni dimostrano che ciascuna delle relazioni coinvolte nel ciclo è essenziale, poiché la loro rimozione comporterebbe la perdita di informazioni cruciali.

### **3.4 Rimozione dei costrutti**

Per proseguire con la progettazione logica, è necessario rimuovere i costrutti dello schema concettuale non esprimibili in uno schema logico, nel nostro caso nel modello relazionale. Alcuni costrutti presi in particolare esame in questo processo sono

#### **3.4.1 Generalizzazioni**

Lo schema E-R finora realizzato non presenta alcun tipo di generalizzazione quindi non è necessario ristrutturarlo.

#### **3.4.2 Attributi multivalore**

Nell'entità FORNITORE, è presente un attributo multivalore *Numero di Telefono*. Lo possiamo rimuovere introducendo una nuova entità TELEFONO e stabilendo una nuova relazione 'Ha' tra questa entità e FORNITORE.

#### **3.4.3 Partizionamenti**

L'entità RESPONSABILE potrebbe essere rappresentata come attributo composto all'interno dell'entità DIPARTIMENTO. Tenere le due entità separate però garantisce l'indipendenza dell'entità e il mantenimento della normalizzazione in quanto un responsabile può gestire più dipartimenti.

#### **3.4.4 Relazioni n-arie**

Il modello realizzato si basa solo su relazioni binarie, non si presenta quindi la necessità di rimuovere alcun tipo di relazione.



### 3.4.5 Scelta delle chiavi primarie

L'entità FORNITORE ha due chiavi primarie candidate in *Partita\_IVA* e *ID\_Fornitore*; per effettuare il passaggio allo schema logico è necessario stabilire quale di queste dovrà identificare l'entità FORNITORE. In questo caso, entrambe le chiavi portano lo stesso significato e non essendo FORNITORE un'entità debole siamo liberi di scegliere una tra le due per cui la scelta è ricaduta su *ID\_ORDINE*.

L'entità RICHIESTA D'ACQUISTO, essendo una entità debole verrà identificata dalla coppia <Numero Progressivo, ORDINE>, dove con ORDINE si fa riferimento all'identificativo di tale entità.

Ogni altra entità ha una sola chiave candidata perciò tale diventerà la chiave primaria nello schema logico.

### 3.5 Schema E-R ristrutturato

Vincoli d'integrità:

- Tutti gli articoli devono essere associati ad un fornitore
- Le richieste d'acquisto di un ordine devono essere soddisfatte tutte da un fornitore.

Vincoli di derivazione:

- *Numero Articoli* si ricava come conteggio degli articoli presenti in una richiesta.  
Deriva dalla relazione Formato da

Lo schema E-R che tiene conto di tutte queste considerazioni è riportato in Figura 8

### 3.6 Schema Logico

#### 3.6.1 Modello relazionale

Legenda:

- Chiavi primarie (PK): Sottolineate
- Chiavi esterne (FK): Corsivo

RESPONSABILE (CF, Nome, Cognome, Secondo Nome(opzionale), Data di Nascita, Sesso)

DIPARTIMENTO (ID\_Dipartimento, *CF\_responsabile*, Nome, Ambito, Descrizione, Sede)

RICHIESTA D'ACQUISTO (Numero Progressivo, ID\_dipartimento, ID\_Ordine, Data, NumeroArticoli)

ARTICOLO (ID\_Articolo, Nome, Descrizione, Unità\_di\_misura, Classe\_Merceologica)

FORMATA\_DA (ID\_dipartimento, Numero\_Richiesta, ID\_Articolo, Quantità\_Articolo, Data\_Prevista\_Consegna)

FORNITORE (ID\_Fornitore, Partita\_IVA, Indirizzo, Email)

TELEFONO (Numero, *Partita\_IVA*)

POSSIEDE (Partita\_IVA, ID\_Articolo, Prezzo\_Unitario, Quantitativo\_Minimo\_Ordine, Sconto)

ORDINE (ID\_Ordine, *Partita\_IVA*, Data\_Emissione, Prezzo\_Ordine)

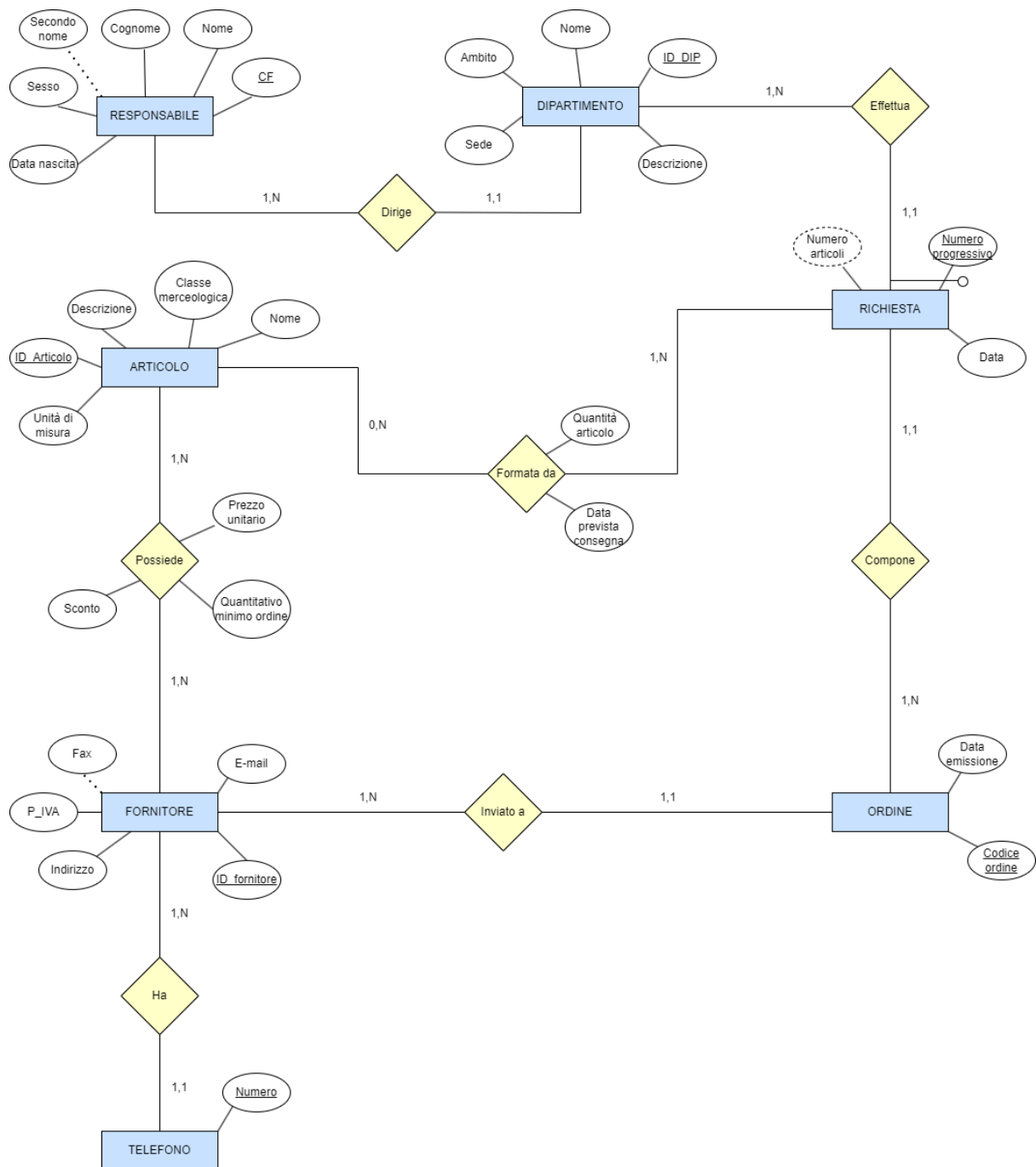


Figura 8: Schema concettuale del modello Entità-Relazioni ristrutturato

### 3.6.2 Vincoli

Vincolo di Verifica{RESPONSABILE.Sesso}

Vincolo Not Null{ORDINE.ID\_ordine o ORDINE.Partita\_IVA}

*CF\_responsabile*: FK di RESPONSABILE rispetto a DIPARTIMENTO

*ID\_Dipartimento*: FK di DIPARTIMENTO rispetto a RICHIESTA D'ACQUISTO

*ID\_Ordine*: FK di ORDINE rispetto a RICHIESTA D'ACQUISTO

*ID\_Dipartimento*: FK di DIPARTIMENTO rispetto a FORMATA\_DA

*ID\_Articolo*: FK di ARTICOLO rispetto a POSSIEDE

*Partita\_IVA*: FK di FORNITORE rispetto a TELEFONO

*Partita\_IVA*: FK di FORNITORE rispetto a POSSIEDE

*ID\_Articolo*: FK di ARTICOLO rispetto a POSSIEDE

*Partita\_IVA*: FK di FORNITORE rispetto a ORDINE

# 4 Progettazione Fisica

## 4.1 Definizioni delle relazioni in SQL

Per la creazione della base di dati si è scelto di svolgere le varie operazioni e commit manualmente attraverso il tool pgAdmin 4.

Per prima cosa è stato creato il database “UfficioAcquisti” e lo schema “ufficio\_acquisti” che conterranno tutti gli elementi necessari. Il codice utilizzato per realizzare queste prime due fasi è il seguente:

```
create database UfficioAcquisti;  
create schema ufficio_acquisti;  
SET search_path TO ufficio_acquisti;
```

Figura 9: codice per la creazione del database e del relativo schema

Successivamente alla creazione della base di dati si è passati alla creazione delle tabelle seguendo i vincoli imposti dal testo e quelli trovati durante la progettazione concettuale e logica.

```
CREATE TABLE Responsabile  
(  
    CF char(16) NOT NULL,      //codice fiscale ha sempre 16 caratteri  
    Nome varchar(50),  
    Cognome varchar(50),  
    Secondo_nome varchar(50),  
    Data_di_nascita date,  
    Sesso char(1), //No alle distinzioni  
    CONSTRAINT responsabile_pkey PRIMARY KEY (CF),  
    CONSTRAINT responsabile_sesso_check CHECK (sesso IN ('M','F'))  
    //controllo che in sesso ci sia o la m o la f  
);
```

Figura 10: codice per la creazione della tabella Responsabile

```
CREATE TABLE Dipartimento
(
  ID_dip varchar(8) NOT NULL,
  CF_resp char(16),
  Nome varchar(50),
  Ambito varchar(50),
  Sede varchar(50),
  Descrizione varchar(200),
  CONSTRAINT dipartimento_pkey PRIMARY KEY (ID_dip),
  CONSTRAINT responsabile_fkey FOREIGN KEY (CF_resp)
REFERENCES    ufficio_acquisti.Responsabile(CF)
);
```

Figura 11: codice per la creazione della tabella Dipartimento

```
CREATE TABLE Articolo
(
  ID_articolo varchar(32) NOT NULL,
  Nome varchar(50),
  Descrizione varchar(200),
  Unità_di_misura varchar(10),
  Classe_merceologica varchar(20),
  CONSTRAINT articolo_pkey PRIMARY KEY
(ID_articolo)
);
```

Figura 12: codice per la creazione della tabella Articolo

```
CREATE TABLE Telefono(
  Numero varchar(15) NOT NULL,
  Partita_iva char(11) NOT NULL,
  CONSTRAINT telefono_pkey PRIMARY KEY (Numero),
  CONSTRAINT fornitore_fkey FOREIGN KEY (Partita_iva)
REFERENCES ufficio_acquisti.Fornitore(Partita_iva)
);
```

Figura 13: codice per la creazione della tabella Telefono, si assume la possibilità di accettare numeri internazionali senza estensioni o altre informazioni

```
CREATE TABLE Fornitore
(
  ID_fornitore varchar(16) NOT NULL,
  Partita_iva char(11) NOT NULL,
  Indirizzo varchar(50),
  Email varchar(20),
  Fax varchar(30),
  CONSTRAINT fornitore_pkey PRIMARY KEY
  (Partita_iva)
);
```

Figura 14: codice per la creazione della tabella Fornitore

```
CREATE TABLE Ordine
(
  ID_ordine varchar(10) NOT NULL,
  Partita_iva char(11) NOT NULL,
  Data_emissione date,
  Prezzo_ordine numeric(10, 2),
  CONSTRAINT identificatore_check CHECK (ID_ordine OR Partita_iva is NOT
  NULL),
  CHECK (ID_ordine OR Partita_iva is NOT NULL),
  CONSTRAINT ordine_pkey PRIMARY KEY (ID_ordine),
  CONSTRAINT ordine_fkey FOREIGN KEY (Partita_iva) REFERENCES
  ufficio_acquisti.Fornitore(Partita_iva)
```

Figura 15: codice per la creazione della tabella Ordine

```
CREATE TABLE Richiesta_Acquisto
(
  Numero_progressivo char(16) NOT NULL,
  ID_dip varchar(8) NOT NULL,
  ID_ordine varchar(10),
  Data_richiesta date,
  N_articoli integer,
  CONSTRAINT richiesta_acquisto_pkey PRIMARY KEY
  (Numero_progressivo, ID_dip),
  CONSTRAINT dipartimento_fkey FOREIGN KEY (ID_dip)
  REFERENCES ufficio_acquisti.Dipartimento(ID_dip),
  CONSTRAINT ordine_fkey FOREIGN KEY (ID_ordine)
  REFERENCES ufficio_acquisti.Ordine(ID_ordine)
);
```

Figura 16: codice per la creazione della tabella Richiesta\_Acquisto

```
CREATE TABLE Possiede
(
  ID_art varchar(32) NOT NULL,
  Partita_iva char(11) NOT NULL,
  Prezzo_unitario numeric(10, 2) NOT NULL,
  Quantitativo_minimo_ordine varchar(10) NOT
  NULL,
  Sconto varchar(3), //espresso in percentuale
  CONSTRAINT possiede_pkey PRIMARY KEY (ID_art,
  Partita_iva),
  CONSTRAINT possiede_fkey FOREIGN KEY
  (Partita_Iva) REFERENCES
  ufficio_acquisti.Fornitore(Partita_Iva),
  CONSTRAINT possiede_fkey2 FOREIGN KEY (ID_art)
  REFERENCES ufficio_acquisti.Articolo(ID_artico
  lo);
```

Figura 17: codice per la creazione della tabella Possiede

```
CREATE TABLE Formata_Da
(
  Numero_Richiesta varchar(16) NOT NULL,
  ID_dip varchar(8) NOT NULL,
  ID_Articolo varchar(32) NOT NULL,
  Quantità_Articolo varchar(10),
  Data_Prevista_Consegna date,
  CONSTRAINT formata_da_pkey PRIMARY KEY (Numero_Richiesta, ID_Articolo,
  ID_dip),
  CONSTRAINT formata_da_fkey FOREIGN KEY (ID_dip, Numero_Richiesta)
  REFERENCES ufficio_acquisti.Richiesta_Acquisto(ID_dip, Numero_progressivo),
  CONSTRAINT formata_da_fkey2 FOREIGN KEY (ID_Articolo) REFERENCES
  ufficio_acquisti.Articolo(ID_articolo)
);
```

Figura 18: codice per la creazione della tabella Formata\_Da

## 4.2 Analisi e scelta degli indici

Quando si introducono indici all'interno del proprio database, si introduce una forma di ridondanza. Questo avviene poiché le stesse informazioni presenti nelle tabelle vengono riorganizzate in altre strutture separate. Pertanto, prima di creare indici per attributi specifici, è necessario valutarne l'efficacia.

Si dovrebbero prendere in considerazione gli attributi più frequentemente utilizzati nelle operazioni all'interno del database. Ad esempio, se si sta eseguendo un'operazione di lettura come "Mostrare la data di consegna prevista", è opportuno creare un indice sull'attributo "data di consegna prevista" per accelerare le query di lettura. Questo perché, conoscendo l'ID della richiesta, è possibile individuare rapidamente la richiesta di interesse attraverso l'indice. Gli indici sono organizzati in modo da supportare una ricerca binaria, e il file indice è notevolmente più piccolo del file di dati, rendendo la ricerca binaria nell'indice molto efficiente.

Tuttavia, se si considera un'operazione come "Cambiare il prezzo unitario di un articolo posseduto da un fornitore", non sarebbe vantaggioso creare un indice per l'attributo "prezzo unitario" nella relazione "Possiede". Ciò perché l'operazione comporta una modifica all'attributo, che richiederebbe una riorganizzazione dell'indice.

In sintesi, l'uso di indici offre vantaggi notevoli in termini di accelerazione delle operazioni di lettura sugli attributi desiderati. D'altra parte, è importante tenere presente che comporta anche svantaggi quando gli attributi vengono modificati, cancellati o ne vengono inseriti di nuovi, poiché ciò richiede la riorganizzazione dell'indice.



## 5. Implementazione

### 5.1 Inserimento dei dati

Si è poi passato all'inserimento dei valori all'interno delle tabelle seguendo un'ordine preciso per evitare di non poter rispettare i vincoli imposti. Il tutto è stato fatto con un'unica transazione.

```
start transaction;
insert into ufficio_acquisti.responsabile(cf, nome, cognome, secondo_nome,
data_di_nascita, sesso) values ('FRGANM89R54N788H', 'GINO', 'PAOLI', 'GINUS',
'1956-04-20', 'M');
insert into ufficio_acquisti.responsabile(cf, nome, cognome, secondo_nome,
data_di_nascita, sesso) values ('GGGANM89R54N788H', 'MARIO', 'ROSSO', 'MAIRS',
'1966-04-25', 'M');
insert into ufficio_acquisti.responsabile(cf, nome, cognome, secondo_nome,
data_di_nascita, sesso) values ('EGGANM89R54N788H', 'GIACOMO', 'VANESSI', 'JACK',
'2007-01-25', 'M');
insert into ufficio_acquisti.responsabile(cf, nome, cognome, secondo_nome,
data_di_nascita, sesso) values ('FHGANM89R54N788H', 'LARA', 'LARINI', 'LAR',
'1999-10-25', 'F');
insert into ufficio_acquisti.responsabile(cf, nome, cognome, secondo_nome,
data_di_nascita, sesso) values ('GGGANM89G74N788H', 'VANIA', 'GRAFIA', 'VANIS',
'1980-05-29', 'F');

insert into ufficio_acquisti.dipartimento(id_dip, cf_resp, nome, ambito, sede, descrizione)
values ('01', 'GGGANM89R54N788H', 'Dipartimento Galileo', 'Ingegneria', 'Milano'
,'Dipartimento dove si studia l'ingegneria delle mele');
insert into ufficio_acquisti.dipartimento(id_dip, cf_resp, nome, ambito, sede, descrizione)
values ('02', 'GGGANM89R54N788H', 'Dipartimento Newton', 'Anatomia', 'Monza'
,'Dipartimento dove si studia l'anatomia delle cicale');
insert into ufficio_acquisti.dipartimento(id_dip, cf_resp, nome, ambito, sede, descrizione)
values ('03', 'FHGANM89R54N788H', 'Dipartimento Bohr', 'Architettura', 'Milano'
,'Dipartimento dove si studia come sono fatti i grattacieli giapponesi');
insert into ufficio_acquisti.dipartimento(id_dip, cf_resp, nome, ambito, sede, descrizione)
values ('04', 'EGGANM89R54N788H', 'Dipartimento Boole', 'Chimica', 'Vicenza'
,'Dipartimento dove si studia le reazioni nucleari');

insert into ufficio_acquisti.articolo(id_articolo, nome, descrizione, unità_di_misura,
classe_merceologica) values ('01', 'martello', 'utensile usato per battere chiodi', 'kg',
'attrezzi fai-da-te');
```

```
insert into ufficio_acquisti.articolo(id_articolo, nome, descrizione, unità_di_misura,
classe_merceologica) values ('02', 'chiodo', 'utensile usato per fissare oggetti', 'kg', 'attrezzi
fai-da-te');
```

```
insert into ufficio_acquisti.articolo(id_articolo, nome, descrizione, unità_di_misura,
classe_merceologica) values ('03', 'becker', 'contenitore per liquidi', 'L', 'contenitori');
```

```
insert into ufficio_acquisti.articolo(id_articolo, nome, descrizione, unità_di_misura,
classe_merceologica) values ('04', 'telescopio', 'oggetto usato per la visione di oggetti
lontani', 'kg', 'astronomia e lenti');
```

```
insert into ufficio_acquisti.articolo(id_articolo, nome, descrizione, unità_di_misura,
classe_merceologica) values ('05', 'stetoscopio', 'utensile per sentire rumori toracici', 'kg',
'medicina');
```

```
insert into ufficio_acquisti.articolo(id_articolo, nome, descrizione, unità_di_misura,
classe_merceologica) values ('06', 'tachipirina', 'medicina per mal di testa', 'g', 'medicina');
```

```
insert into ufficio_acquisti.telefono(numero, partita_iva) values ('3899088902',
'12458614935');
```

```
insert into ufficio_acquisti.telefono(numero, partita_iva) values ('3899388902',
'72576483124');
```

```
insert into ufficio_acquisti.telefono(numero, partita_iva) values ('4899088907',
'24751023678');
```

```
insert into ufficio_acquisti.telefono(numero, partita_iva) values ('6899088907',
'57814963687');
```

```
insert into ufficio_acquisti.telefono(numero, partita_iva) values ('3899077907',
'72576483124');
```

```
insert into ufficio_acquisti.telefono(numero, partita_iva) values ('3894322907',
'91201248967');
```

```
insert into ufficio_acquisti.telefono(numero, partita_iva) values ('3439088909',
'61254789648');
```

```
insert into ufficio_acquisti.telefono(numero, partita_iva) values ('3810088907',
'48560124657');
```

```
insert into ufficio_acquisti.telefono(numero, partita_iva) values ('3429088908',
'84596789301');
```

```
insert into ufficio_acquisti.telefono(numero, partita_iva) values ('7899088907',
'24570124954');
```

```
insert into ufficio_acquisti.fornitore(id_fornitore, partita_iva, indirizzo, email, fax) values
('F00001', '12458614935', 'Via Roma 16', 'pittini@gmail.com', '012564891457@fax.tc');
```

```
insert into ufficio_acquisti.fornitore(id_fornitore, partita_iva, indirizzo, email) values
('F00002', '72576483124', 'Via Milano 15', '456178345168@fax.tc');
```

```
insert into ufficio_acquisti.fornitore(id_fornitore, partita_iva, indirizzo, email) values
('F00003', '24751023678', 'Via Torino 16', 'fantoni@gmail.com');
```

```
insert into ufficio_acquisti.fornitore(id_fornitore, partita_iva, indirizzo, fax) values ('F00004',
'57814963687', 'Via Udine 16', '012564891457@fax.tc');
```

```
insert into ufficio_acquisti.fornitore(id_fornitore, partita_iva, indirizzo, email, fax) values
('F00005', '91201248967', 'Via Trieste 16', 'danieli@gmail.com', '2457316485996@fax.tc');
insert into ufficio_acquisti.fornitore(id_fornitore, partita_iva, indirizzo, fax) values ('F00006',
'61254789648', 'Via Napoli 16', '124578316485@fax.tc');
insert into ufficio_acquisti.fornitore(id_fornitore, partita_iva, indirizzo, email) values
('F00007', '48560124657', 'Via Bologna 16', 'mondadori@gmail.com');
insert into ufficio_acquisti.fornitore(id_fornitore, partita_iva, indirizzo, fax) values ('F00008',
'84596789301', 'Via Palermo 16', '124576321679@fax.tc');
insert into ufficio_acquisti.fornitore(id_fornitore, partita_iva, indirizzo, fax) values ('F00009',
'24570124954', 'Via Venezia 16', '124657921789@fax.tc');
```

```
insert into ufficio_acquisti.ordine(id_ordine, partita_iva, data_emissione, prezzo_ordine)
values ('01', '12458614935', '2023-01-25', 40.23);
insert into ufficio_acquisti.ordine(id_ordine, partita_iva, data_emissione, prezzo_ordine)
values ('02', '24570124954', '2023-01-28', 100.23);
insert into ufficio_acquisti.ordine(id_ordine, partita_iva, data_emissione, prezzo_ordine)
values ('03', '24570124954', '2023-01-26', 32.27);
insert into ufficio_acquisti.ordine(id_ordine, partita_iva, data_emissione, prezzo_ordine)
values ('04', '48560124657', '2023-01-28', 76.65);
insert into ufficio_acquisti.ordine(id_ordine, partita_iva, data_emissione, prezzo_ordine)
values ('05', '61254789648', '2023-01-30', 109.80);
```

```
insert into ufficio_acquisti.richiesta_acquisto(numero_progressivo, id_dip, id_ordine,
data_richiesta, n_articoli) values ('1', '01', '02', '2023-01-10', 57);
insert into ufficio_acquisti.richiesta_acquisto(numero_progressivo, id_dip, id_ordine,
data_richiesta, n_articoli) values ('2', '03', '02', '2023-01-11', 21);
insert into ufficio_acquisti.richiesta_acquisto(numero_progressivo, id_dip, id_ordine,
data_richiesta, n_articoli) values ('3', '02', '03', '2023-01-10', 4);
insert into ufficio_acquisti.richiesta_acquisto(numero_progressivo, id_dip, id_ordine,
data_richiesta, n_articoli) values ('4', '04', '05', '2023-01-15', 50);
insert into ufficio_acquisti.richiesta_acquisto(numero_progressivo, id_dip, id_ordine,
data_richiesta, n_articoli) values ('5', '02', '01', '2023-01-20', 90);
insert into ufficio_acquisti.richiesta_acquisto(numero_progressivo, id_dip, id_ordine,
data_richiesta, n_articoli) values ('6', '01', '03', '2023-01-14', 290);
insert into ufficio_acquisti.richiesta_acquisto(numero_progressivo, id_dip, id_ordine,
data_richiesta, n_articoli) values ('7', '04', '01', '2023-01-14', 32);
```

```
insert into ufficio_acquisti.possiede(id_art, partita_iva, prezzo_unitario,
quantitativo_minimo_ordine, sconto) values ('03', '12458614935', 18.20, '3', '3%');
insert into ufficio_acquisti.possiede(id_art, partita_iva, prezzo_unitario,
quantitativo_minimo_ordine) values ('05', '12458614935', 47.50, '6');
```

```
insert into ufficio_acquisti.possiede(id_art, partita_iva, prezzo_unitario,
quantitativo_minimo_ordine, sconto) values ('01', '72576483124', 12.00, '3', '2%');
insert into ufficio_acquisti.possiede(id_art, partita_iva, prezzo_unitario,
quantitativo_minimo_ordine) values ('02', '72576483124', 1.20, '15');
insert into ufficio_acquisti.possiede(id_art, partita_iva, prezzo_unitario,
quantitativo_minimo_ordine) values ('04', '72576483124', 250.00, '5');
insert into ufficio_acquisti.possiede(id_art, partita_iva, prezzo_unitario,
quantitativo_minimo_ordine, sconto) values ('01', '24751023678', 12.20, '5', '2%');
insert into ufficio_acquisti.possiede(id_art, partita_iva, prezzo_unitario,
quantitativo_minimo_ordine, sconto) values ('02', '24751023678', 1.30, '6', '3%');
insert into ufficio_acquisti.possiede(id_art, partita_iva, prezzo_unitario,
quantitativo_minimo_ordine) values ('04', '24751023678', 245.50, '2');
insert into ufficio_acquisti.possiede(id_art, partita_iva, prezzo_unitario,
quantitativo_minimo_ordine, sconto) values ('03', '57814963687', 19.00, '4', '2%');
insert into ufficio_acquisti.possiede(id_art, partita_iva, prezzo_unitario,
quantitativo_minimo_ordine) values ('04', '57814963687', 210.20, '1');
insert into ufficio_acquisti.possiede(id_art, partita_iva, prezzo_unitario,
quantitativo_minimo_ordine, sconto) values ('04', '91201248967', 260.00, '2', '2%');
insert into ufficio_acquisti.possiede(id_art, partita_iva, prezzo_unitario,
quantitativo_minimo_ordine, sconto) values ('03', '91201248967', 19.00, '4', '3%');
insert into ufficio_acquisti.possiede(id_art, partita_iva, prezzo_unitario,
quantitativo_minimo_ordine) values ('05', '61254789648', 48.50, '6');
insert into ufficio_acquisti.possiede(id_art, partita_iva, prezzo_unitario,
quantitativo_minimo_ordine, sconto) values ('01', '61254789648', 17.00, '2', '3%');
insert into ufficio_acquisti.possiede(id_art, partita_iva, prezzo_unitario,
quantitativo_minimo_ordine) values ('02', '61254789648', 1.30, '10');
insert into ufficio_acquisti.possiede(id_art, partita_iva, prezzo_unitario,
quantitativo_minimo_ordine) values ('04', '48560124657', 290.00, '2');
insert into ufficio_acquisti.possiede(id_art, partita_iva, prezzo_unitario,
quantitativo_minimo_ordine, sconto) values ('01', '48560124657', 12.10, '7', '10%');
insert into ufficio_acquisti.possiede(id_art, partita_iva, prezzo_unitario,
quantitativo_minimo_ordine, sconto) values ('02', '48560124657', 1.15, '10', '3%');
insert into ufficio_acquisti.possiede(id_art, partita_iva, prezzo_unitario,
quantitativo_minimo_ordine, sconto) values ('03', '84596789301', 18.75, '6', '12%');
insert into ufficio_acquisti.possiede(id_art, partita_iva, prezzo_unitario,
quantitativo_minimo_ordine) values ('04', '84596789301', 226.20, '2');
insert into ufficio_acquisti.possiede(id_art, partita_iva, prezzo_unitario,
quantitativo_minimo_ordine, sconto) values ('03', '24570124954', 19.75, '8', '11%');
insert into ufficio_acquisti.possiede(id_art, partita_iva, prezzo_unitario,
quantitativo_minimo_ordine) values ('04', '24570124954', 242.80, '1');
```

```
insert into ufficio_acquisti.formata_da(numero_richiesta, id_dip, id_articolo,
quantità_articolo, data_prevista_consegna) values ('2', '03', '06', '40', '2023-03-10');
insert into ufficio_acquisti.formata_da(numero_richiesta, id_dip, id_articolo,
quantità_articolo, data_prevista_consegna) values ('4', '04', '06', '19', '2023-03-13');
insert into ufficio_acquisti.formata_da(numero_richiesta, id_dip, id_articolo,
quantità_articolo, data_prevista_consegna) values ('7', '04', '02', '22', '2023-03-29');
insert into ufficio_acquisti.formata_da(numero_richiesta, id_dip, id_articolo,
quantità_articolo, data_prevista_consegna) values ('6', '01', '05', '7', '2023-03-01');
insert into ufficio_acquisti.formata_da(numero_richiesta, id_dip, id_articolo,
quantità_articolo, data_prevista_consegna) values ('5', '02', '04', '15', '2023-04-10');
insert into ufficio_acquisti.formata_da(numero_richiesta, id_dip, id_articolo,
quantità_articolo, data_prevista_consegna) values ('2', '03', '01', '7', '2023-05-01');
insert into ufficio_acquisti.formata_da(numero_richiesta, id_dip, id_articolo,
quantità_articolo, data_prevista_consegna) values ('5', '02', '01', '15', '2023-04-13');
insert into ufficio_acquisti.formata_da(numero_richiesta, id_dip, id_articolo,
quantità_articolo, data_prevista_consegna) values ('1', '01', '04', '15', '2023-05-10');
insert into ufficio_acquisti.formata_da(numero_richiesta, id_dip, id_articolo,
quantità_articolo, data_prevista_consegna) values ('4', '04', '02', '22', '2023-03-29');
insert into ufficio_acquisti.formata_da(numero_richiesta, id_dip, id_articolo,
quantità_articolo, data_prevista_consegna) values ('7', '04', '01', '7', '2023-02-01');

commit;
```

## 5.2 Definizione dei Trigger

Un trigger in una base di dati è una procedura o un insieme di istruzioni predefinite che vengono attivate automaticamente quando si verifica un determinato evento o condizione all'interno del database. Gli eventi che possono scatenare un trigger includono l'inserimento, la modifica o la cancellazione di dati in una tabella. I trigger vengono utilizzati per eseguire azioni specifiche, come il controllo dei dati prima che vengano inseriti o aggiornati, l'automazione di compiti come l'aggiornamento di altre tabelle o l'invio di notifiche. In sostanza, i trigger consentono di automatizzare e personalizzare il comportamento del database in risposta agli eventi che si verificano al suo interno.

Si è scelto di implementare 2 trigger, uno per il controllo di un valore ed uno per rispettare un vincolo del problema, i trigger che sono stati definiti sono i seguenti:

1. Trigger per controllare che il prezzo di un articolo sia un valore positivo;
2. Trigger per evitare la cancellazione di un responsabile che controlla uno o più dipartimenti

### 5.2.1 Prezzo articolo

Il seguente trigger chiamato “before\_insert\_on\_possiede” verifica il valore del prezzo di un prodotto durante il suo inserimento o il suo aggiornamento. Viene eseguito per ogni riga attraverso la funzione ausiliaria “verifica\_prezzo” che controlla tale valore e solleva un’eccezione nel caso in cui sia minore o uguale a zero stampando un messaggio di errore. Se invece il prezzo è valido l’operazione viene eseguita con successo.

```
CREATE OR REPLACE FUNCTION controlla_prezzo()
RETURNS TRIGGER LANGUAGE PLPGSQL AS
$$
DECLARE prezzo FLOAT;
BEGIN
    SELECT Prezzo_unitario INTO prezzo FROM
    ufficio_acquisti.possiede;
    IF prezzo <= 0 THEN
        RAISE NOTICE 'Prezzo non valido';
        RETURN NULL;
    END IF;
    RETURN NEW;
END;
$$;

CREATE TRIGGER 'before_insert_on_possiede'
BEFORE UPDATE OR INSERT ON ufficio_acquisti.possiede
FOR EACH ROW
EXECUTE PROCEDURE controlla_prezzo();
```

Figura 19: codice per l’implementazione del trigger

### 5.2.2 Cancellazione responsabile

Il seguente trigger chiamato “controlla\_responsabile\_dipartimento”, viene eseguito in fase di cancellazione di un responsabile, tale azione non può essere infatti eseguita se il responsabile controlla uno o più dipartimenti.

Per fare ciò viene utilizzata la funzione ausiliaria “controlla\_responsabile” che viene eseguita per ogni riga. Nello specifico tale funzione conta il numero di dipartimenti associati al codice fiscale del responsabile, se tale valore è diverso da zero viene sollevata un’eccezione e stampato il numero di dipartimenti che controlla, altrimenti l’operazione ha successo.

```
CREATE OR REPLACE FUNCTION controlla_responsabile()
RETURNS TRIGGER LANGUAGE PLPGSQL AS
$$
DECLARE ndip INT;
DECLARE log_message TEXT;
BEGIN
    SELECT COUNT(*) INTO ndip
    FROM ufficio_acquisti.dipartimento, ufficio_acquisti.responsabile
    WHERE responsabile.cf = dipartimento.cf_resp;
    IF ndip <> 0 THEN
        log_message := CONCAT('Il responsabile controlla ', ndip, '
dipartimento/i, quindi non può essere cancellato');
        RAISE NOTICE '%', log_message;
        RETURN NULL;
    END IF;
    RETURN NEW;
END;
$$;

CREATE TRIGGER 'controlla_responsabile_dipartimento'
BEFORE DELETE ON ufficio_acquisti.responsabile
FOR EACH ROW
EXECUTE PROCEDURE controlla_responsabile();
```

Figura 20: codice per l'implementazione del trigger

### 5.3 Definizione di Query

Per la definizione delle query si è scelto di implementare le tre operazioni più interessanti tra quelle definite nella sezione 1.2

#### Query 1

La prima query corrisponde all'operazione 1 ovvero interrogare la richiesta d'acquisto per sapere il numero di articoli che contiene. L'informazione che si sta cercando è completamente contenuta nella tabella "richiesta\_acquisto".

```
SELECT n_articoli
FROM ufficio_acquisti.richiesta_acquisto;
```

Figura 21: codice della query n.1

#### Query 2

La seconda query corrisponde all'operazione 10 ovvero mostrare la data prevista di consegna. Anche in questo caso l'informazione ricercata è contenuta in un'unica tabella, nello specifico nella tabella "formata da".

```
SELECT data_prevista_consegna  
FROM ufficio_acquisti.formata_da;
```

Figura 22: codice della query n.2

### Query 3

La terza e ultima query che si è scelto di definire corrisponde all'operazione 5, ovvero mostrare i dipartimenti controllati da un responsabile. In questo caso l'informazione non è più contenuta in una sola tabella ma in due, ovvero "dipartimento" e "responsabile".

```
SELECT cf, resp.nome, cognome, id_dip  
FROM ufficio_acquisti.dipartimento dip,  
      ufficio_acquisti.responsabile resp  
WHERE dip.cf_resp = resp.cf;
```

Figura 23: codice della query n.3



## 6 Analisi dei dati in R

### 6.1 Numero di fornitori che forniscono un dato articolo

L'obiettivo di questa analisi è la visualizzazione del numero di fornitori che offrono un determinato articolo. Abbiamo iniziato eseguendo una query sul database per ottenere i dati necessari. La query ha prodotto un file CSV contenente le informazioni richieste.

```
select art.nome, count(*) as numeroVenditori
from ufficio_acquisti.articolo as art, ufficio_acquisti.fornitore as
forn, ufficio_acquisti.possiede as pos
where pos.id_art = art.id_articolo and pos.partita_iva = forn.partita_iva
group by art.nome
```

Figura 24: query per il primo grafico

Successivamente, utilizzando il linguaggio R, abbiamo creato un grafico a barre con l'asse X rappresentante gli articoli e l'asse Y rappresentante il numero di fornitori. Questo grafico ci ha permesso di identificare l'articolo più diffuso tra i fornitori (il telescopio, con 7 fornitori) e quello meno diffuso (lo stetoscopio, disponibile solo da due fornitori).

```
NumFornitoriPerArticolo <- read.csv("NumFornitoriArticolo.csv")
H <- c(NumFornitoriPerArticolo$numerovenditori)
M <- c(NumFornitoriPerArticolo$nome)
barplot(H,names.arg=M,main="Fornitori per
articolo",xlab="Articolo",ylab="Numero Fornitori", col="blue")
```

Figura 25: codice del primo grafico

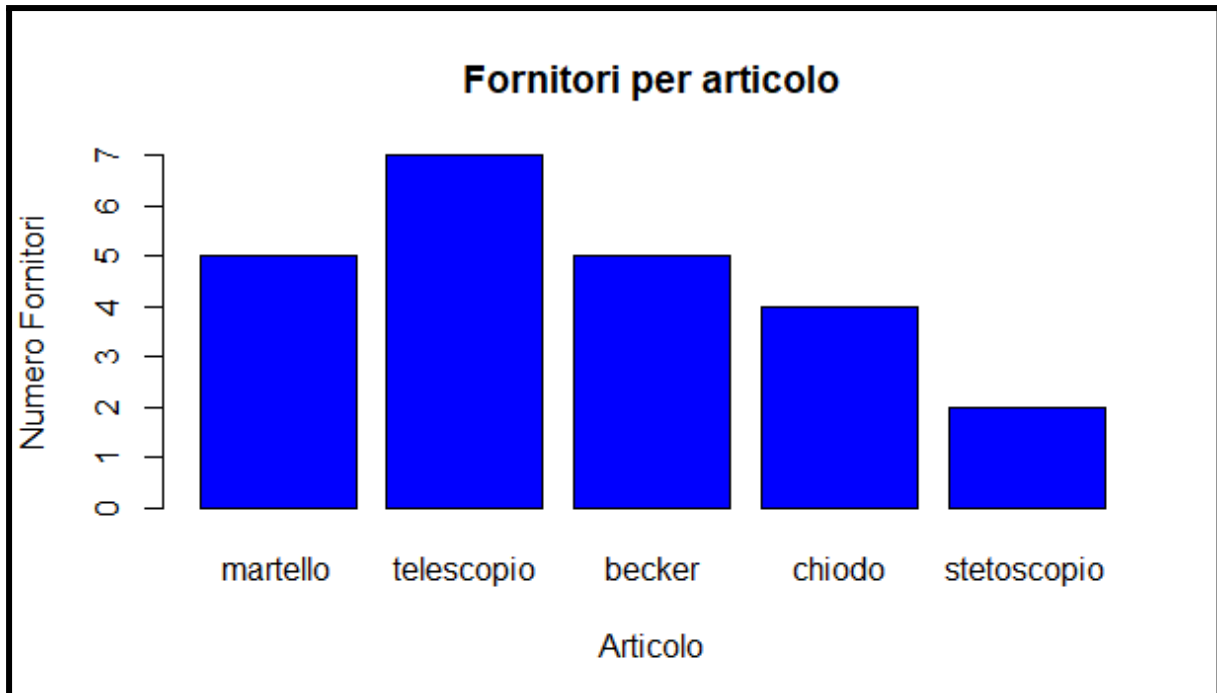


Figura 26: Numero di fornitori per articolo

## 6.2 Totale articoli richiesti per dipartimento

In questa parte dell'analisi, l'obiettivo era calcolare il numero totale di articoli richiesti da ciascun dipartimento. Abbiamo realizzato ciò combinando i dati provenienti da due tabelle diverse tramite la funzione JOIN in una query SQL. La query ha restituito un elenco di dipartimenti con la somma totale degli articoli richiesti.

```
select dip.id_dip, sum(ric.n_articoli) as tot_articoli
from ufficio_acquisti.dipartimento as dip
join ufficio_acquisti.richiesta_acquisto as ric on dip.id_dip = ric.id_dip
group by dip.id_dip
```

Figura 27: query per il secondo grafico

Successivamente, nel linguaggio R, abbiamo creato un grafico a barre in cui sull'asse X sono stati rappresentati i dipartimenti e sull'asse Y il numero di articoli. Questo grafico ci ha permesso di osservare che il dipartimento numero 1 ha effettuato significativamente più ordini rispetto agli altri.

```
dipNumArticoli <- read.csv("dip_narticoli.csv")
H <- c(dipNumArticoli$tot_articoli)
M <- c(dipNumArticoli$id_dip)
barplot(H,names.arg=M,main="Articoli richiesti per
dipartimento",xlab="Dipartimento",ylab="Numero articoli",
col="dark green")
```

Figura 28: codice del secondo grafico

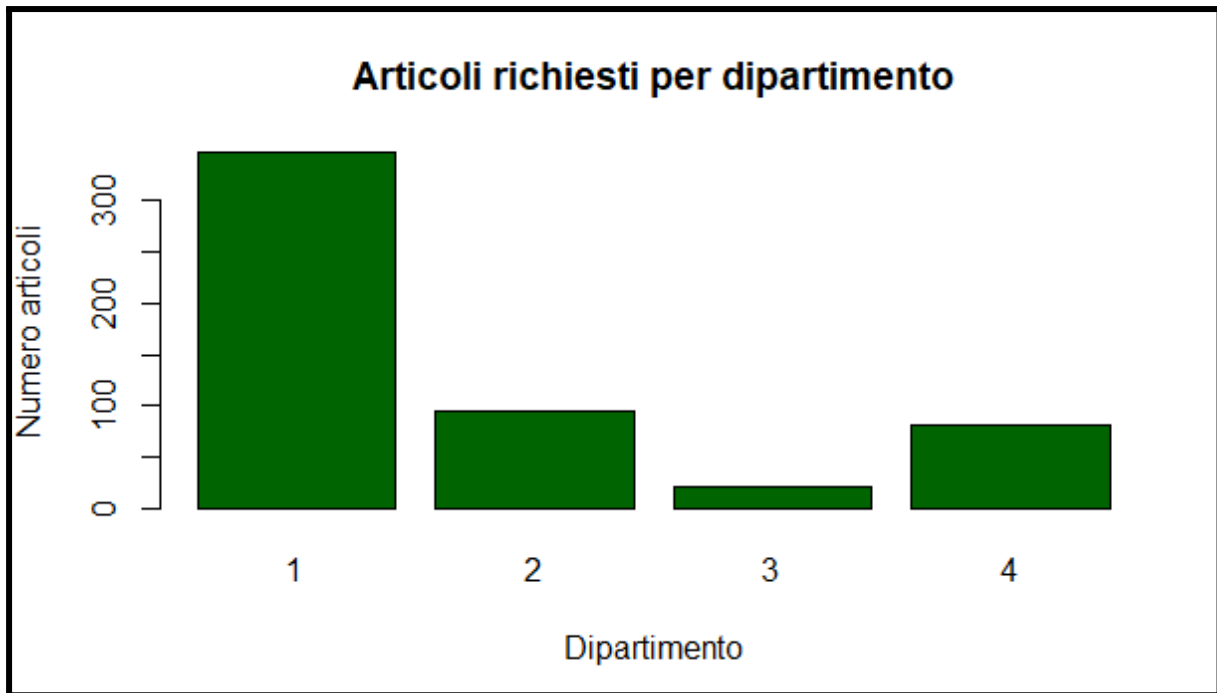


Figura 29: Totale articoli per dipartimento

### 6.3 Prezzo totale delle richieste per dipartimento

In questa fase, l'obiettivo era visualizzare le spese totali derivanti dalle richieste di acquisto per ciascun dipartimento. Abbiamo iniziato con una query SQL per ottenere l'elenco dei dipartimenti con la somma totale dei prezzi degli ordini effettuati da ciascuno di essi.

```
select dip.id_dip, sum(ord.prezzo_ordine) as tot_prezzo
from ufficio_acquisti.dipartimento as dip
join ufficio_acquisti.richiesta_acquisto as ric on dip.id_dip =
ric.id_dip
join ufficio_acquisti.ordine as ord on ric.id_ordine = ord.id_ordine
group by dip.id_dip
```

Figura 30: codice terza query

Successivamente, utilizzando il linguaggio R, abbiamo creato un grafico a barre in cui sull'asse X sono stati rappresentati i dipartimenti e sull'asse Y il totale delle spese. Questo grafico ha evidenziato che, nonostante il dipartimento numero 1 abbia effettuato un grande numero di ordini, non è quello con la spesa più alta.

```
TotPrezzoRichieste <- read.csv("TotPrezzoRichiesteDip.csv")  
H <- c(TotPrezzoRichieste$tot_prezzo)  
M <- c(TotPrezzoRichieste$id_dip)  
barplot(H,names.arg=M,main="Spese per  
dipartimento",xlab="Dipartimento",ylab="Spesa", col="dark red")
```

Figura 31: codice del terzo grafico

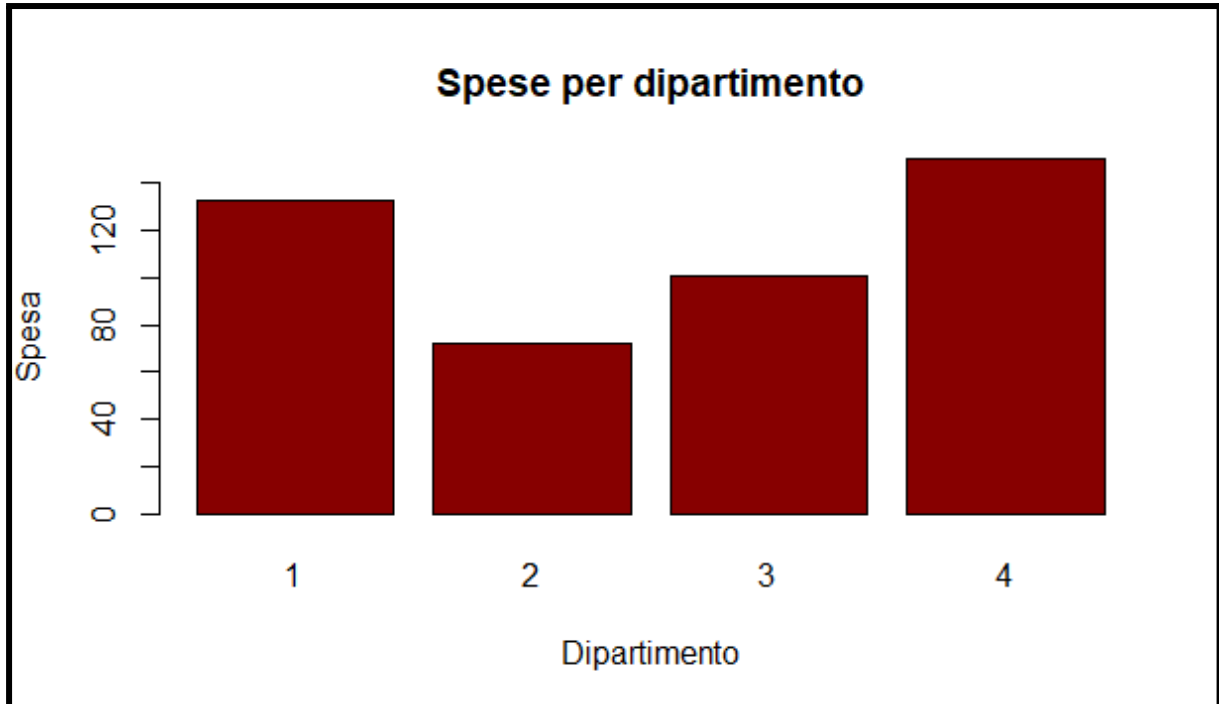


Figura 26: Spese per dipartimento

Questi risultati forniscono una panoramica importante sulle attività di approvvigionamento dei dipartimenti e possono essere utilizzati per prendere decisioni informative in merito alla gestione delle risorse e delle forniture.