

Relazione Progetto Basi di Dati - A.A. 2020-2021

Francesco Bombassei De Bona (144665)

Andrea Cantarutti (141808)

Lorenzo Bellina (142544)

Alessandro Fabris (142520)

11/08/2021

Indice

1	Introduzione	4
2	Analisi dei requisiti	5
2.1	Requisiti	5
2.2	Glossario	6
2.3	Ristesura e strutturazione dei requisiti	7
2.4	Individuazione dei principali requisiti operazionali	8
2.5	Criteri per la rappresentazione dei concetti	9
2.5.1	Assunzioni in merito alle ambiguità rilevate	9
3	Progettazione concettuale	10
3.1	Diagramma ER	10
3.2	Osservazioni	11
3.2.1	Vincoli aziendali	11
3.2.2	Regole di derivazione	11
3.2.3	Considerazioni	11
4	Progettazione logica	12
4.1	Analisi delle ridondanze	12
4.1.1	Analisi dei cicli	12
4.1.2	Attributi derivabili	12
4.1.3	Data di Consegna	13
4.1.4	Numero Articoli	14
4.2	Eliminazione delle generalizzazioni	15
4.3	Partizionamento ed accorpamento di entità e associazioni	15
4.3.1	Reifica di relazioni binarie	15
4.3.2	Reifica delle relazioni ternarie	15
4.3.3	Valutazione degli attributi composti	15
4.3.4	Eliminazione di attributi multivalore	15
4.3.5	Ristrutturazione del diagramma ER	16
4.4	Scelta degli identificatori primari	17
4.5	Traduzione verso il modello logico-relazionale	17
4.5.1	Traduzione di Entità	17
4.5.2	Traduzione di Relazioni Uno a Molti	18
4.5.3	Traduzione di relazioni molti a molti e uno a uno	18
4.5.4	Osservazioni	18
4.6	Modello Relazionale	19
5	Progettazione Fisica	20
5.1	Osservazioni sugli indici	20
5.1.1	Indicizzazione di Include su Ordine in operazioni di ricerca	21
5.1.2	Indicizzazione di Include su Ordine in operazioni di inserimento	22
5.1.3	Indicizzazione di DataEmissione su RichiestaAcquisto in operazioni di ricerca	25
5.1.4	Indicizzazione di DataEmissione su RichiestaAcquisto in operazioni di modifica	26
6	Implementazione	29
6.1	Containerizzazione del DBMS	29
6.2	SQL	29
6.2.1	Definizione dei tipi enum	29
6.2.2	Creazione delle tabelle	30
6.2.3	Definizione dei trigger	33
6.2.4	Definizione degli indici	41
6.3	Produzione ed Inserimento dei dati di Mockup	41
6.4	Generazione della base di dati	41
6.5	Query significative	42

7	Analisi dei dati	47
7.1	Distribuzione delle classi merceologiche	48
7.2	Distribuzione degli articoli per ogni fornitore	49
7.3	Confronto della spesa dei dipartimenti	50
7.4	Spesa totale per classe merceologica	51
7.5	Richieste d'acquisto trimestrali effettuate dai dipartimenti	53
7.6	Numero di richieste d'acquisto mensili	54
7.7	Spesa dei dipartimenti nel mese di giugno	55
7.8	Spesa giornaliera dei dipartimenti	56
8	Conclusioni	57

1 Introduzione

Il presente elaborato espone l'attività di progettazione e implementazione di una Base di Dati relazionale, con una successiva analisi dei dati sperimentali in essa contenuti tramite apposite interrogazioni in linguaggio SQL.

2 Analisi dei requisiti

2.1 Requisiti

La consegna assegnata riporta requisiti il cui **dominio di interesse** è relativo al sistema di gestione dell'*ufficio acquisti di un ente pubblico*.

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 d'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).

Sulla base di quanto riportato, si procede alla formulazione di un glossario che permetta la definizione univoca dei concetti esposti.

2.2 Glossario

La terminologia individuata appartenente al dominio di interesse e correlata alla strutturazione della Base di Dati è presentata di seguito:

Termine	Descrizione	Sinonimi	Relazioni
Dipartimento	Sottosezione organizzativa dell'ente		Responsabile, Richiesta d'acquisto
Responsabile	Persona incaricata delle responsabilità relative ad uno o più dipartimenti		Dipartimento
Richiesta d'acquisto	Documento, formulato da un dipartimento, riportante i riferimenti agli articoli da ordinare, con annesse specifiche	Richiesta	Dipartimento, Articolo
Articolo	Elemento atomico richiedibile ed ordinabile		Richiesta d'acquisto, Listino, Ordine
Fornitore	Azienda che provvede alla fornitura di articoli per l'ente		Listino, Ordine
Listino	Catalogo contenente uno o più articoli relativi ad un fornitore		Articolo, Fornitore
Ordine	Insieme di articoli richiesti dall'ufficio acquisti ad un fornitore per uno o più dipartimenti		Articolo, Fornitore

2.3 Ristesura e strutturazione dei requisiti

A seguito dell'identificazione e organizzazione delle terminologie riportate nel precedente glossario, si identificano e raggruppano le frasi relative a requisiti espressi in linguaggio naturale sulla base di ciò che esse riferiscono.

Dipartimento
<ul style="list-style-type: none">• 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• Ogni dipartimento possa formulare delle richieste d'acquisto
Responsabile
<ul style="list-style-type: none">• Una stessa persona possa essere responsabile di più dipartimenti
Richiesta d'Acquisto
<ul style="list-style-type: none">• Caratterizzata da un numero progressivo, che la identifica univocamente all'interno dell'insieme delle richieste del dipartimento, da una data, dall'insieme degli articoli da ordinare, con l'indicazione, per ciascun articolo, della quantità richiesta, e dalla data prevista di consegna• Si assuma che uno stesso dipartimento possa effettuare più richieste in una stessa data
Articolo
<ul style="list-style-type: none">• 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• 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
Fornitore
<ul style="list-style-type: none">• 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
Listino
<ul style="list-style-type: none">• 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
Ordine
<ul style="list-style-type: none">• 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

2.4 Individuazione dei principali requisiti operazionali

Sulla base dei requisiti individuati, si descrivono le principali operazioni sui dati, con rispettiva frequenza. Si considera, per dare consistenza al conteggio, un ente costituito da trenta dipartimenti e associato a cinque fornitori diversi.

Operazione	Frequenza
Inserimento di una richiesta d'acquisto	60/settimana
Aggiornamento dello stato di un ordine	10/settimana
Visualizzazione delle informazioni relative ad una richiesta d'acquisto	120/settimana
Visualizzazione degli articoli contenuti in una richiesta d'acquisto	180/settimana
Inserimento di un nuovo ordine	5/settimana
Visualizzazione di tutti gli articoli	200/settimana
Calcolo della spesa mensile dei dipartimenti	30/mese
Calcolo della spesa complessiva dell'ente in un intervallo di tempo	5/mese

2.5 Criteri per la rappresentazione dei concetti

Sulla base del documento di specifiche, si individuano i criteri opportuni per la rappresentazione dei concetti descritti.

- 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 d'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** 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 d'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).

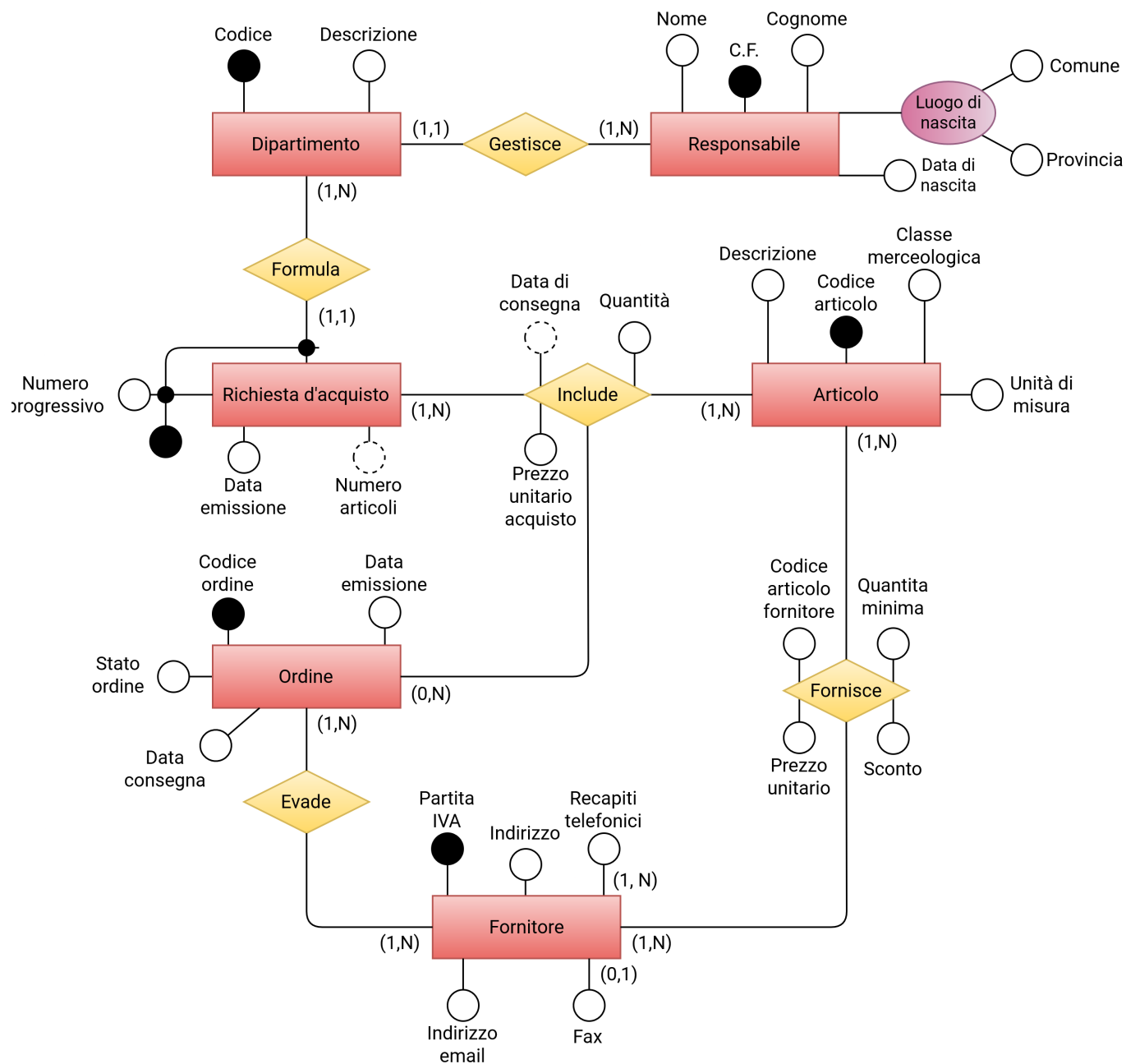
Legenda: **Entità** **Attributo** **Ambiguità** **Relazioni** **Attributi di relazione**

2.5.1 Assunzioni in merito alle ambiguità rilevate

- Sulla base di quanto riportato nelle specifiche sopracitate, si è osservato come il concetto di **listino** delinei l'insieme di articoli associati al rispettivo fornitore senza, però, aggiungere informazioni supplementari in merito a tale relazione. Si è, pertanto, deciso di **non** rappresentare il listino all'interno della Basi di Dati ma di, piuttosto, rappresentare l'associazione fra un singolo articolo e il rispettivo fornitore.
- Si assume che un articolo possa essere fornito da un insieme di fornitori e che, di conseguenza, mentre una richiesta d'acquisto si rivolge agli articoli, è responsabilità dell'ufficio acquisti l'individuazione dello specifico fornitore, in merito ad aspetti logistici e di convenienza.
- Si assume che sia di interesse dell'ente la possibilità di ricondurre un ordine alle richieste d'acquisto che esso soddisfa e una richiesta d'acquisto agli ordini che la coinvolgono.
- Si osserva, inoltre, la necessità di memorizzare il prezzo al quale ogni singolo articolo viene acquistato nell'eventualità che vengano successivamente variati lo sconto e/o il prezzo unitario.
- Infine, sapendo che un ordine coinvolge al più un fornitore e che gli articoli inclusi nelle richieste d'acquisto possono potenzialmente provenire da fornitori diversi si assume che:
 - Un singolo ordine possa soddisfare una richiesta d'acquisto anche parzialmente;
 - Per ogni articolo coinvolto, venga soddisfatta la quantità specificata.

3 Progettazione concettuale

3.1 Diagramma ER



3.2 Osservazioni

Sulla base del diagramma ER proposto, si riportano le osservazioni effettuate, includendo i **vincoli aziendali** individuati e le eventuali **regole di derivazione**.

3.2.1 Vincoli aziendali

Il diagramma presenta un singolo ciclo che coinvolge le entità *Ordine*, *Articolo* e *Fornitore*. Sulla base di quanto riportato nei requisiti si introduce il seguente vincolo aziendale: ***il fornitore degli articoli relativi ad un ordine deve essere il medesimo di quello associato all'ordine stesso.***

Inoltre, si evidenzia come sia la **data di consegna di un articolo** che il **prezzo di acquisto di un articolo** relativamente ad una richiesta, possano essere disponibili solo in seguito alla partecipazione di un ordine alla relazione.

3.2.2 Regole di derivazione

Il diagramma presenta due attributi derivati, ovvero **Data di Consegna** e **Numero Articoli**. Il primo è relativo alla relazione Include e viene derivato sulla base della data di consegna relativa all'ordine che soddisfa ciascun articolo. Il secondo, invece, è relativo all'entità Richiesta d'Acquisto e viene calcolato contando gli articoli associati ad una richiesta (considerandone la rispettiva quantità ordinata).

3.2.3 Considerazioni

Si osserva come la partecipazione dell'entità *Ordine* alla relazione ternaria che coinvolge le entità *Richiesta d'Acquisto*, *Ordine* e *Articolo* sia **opzionale**. Quest'ultima avverrà, infatti, solamente all'atto di emissione (da parte dell'ufficio acquisti) di un ordine che soddisfa l'articolo incluso in una specifica richiesta.

4 Progettazione logica

4.1 Analisi delle ridondanze

4.1.1 Analisi dei cicli

Come specificato precedentemente, l'unico ciclo presente nello schema ER coinvolge le entità **Ordine**, **Articolo** e **Fornitore**. Un ordine contiene degli articoli e viene evaso da uno specifico fornitore. Gli articoli devono essere forniti dal fornitore che evade l'ordine.

Pertanto, il ciclo viene mantenuto e vincolato sulla base delle osservazioni effettuate al punto 3.2.

4.1.2 Attributi derivabili

Al fine di valutare il mantenimento o l'eliminazione delle ridondanze presenti nel diagramma ER proposto, si definisce, di seguito, la tavola dei volumi di entità e relazioni presenti nella Base di Dati. Si considera quanto segue:

- La stato della base di dati dopo un anno di attività
- Richieste d'acquisto che coinvolgono mediamente 5 articoli e soddisfatte da 3 ordini
- Ordini che contengono, in media, 60 articoli
- Ordini che soddisfano mediamente 12 richieste d'acquisto.

Concetto	Tipo	Volume
Responsabile	E	25
Dipartimento	R	30
Richiesta d'Acquisto	E	3120
Articolo	E	300
Ordine	E	260
Fornitore	E	5
Include	R	15600
Fornisce	R	450

Si fa riferimento, inoltre, alle operazioni frequenti riportate al punto 2.4.

Si effettua, quindi, un'analisi delle ridondanze in merito agli attributi derivati **Data di Consegna** della relazione **Include** e **Numero Articoli** dell'entità **Richiesta d'Acquisto**.

Il primo, è coinvolto nelle operazioni di:

- Visualizzazione degli articoli contenuti in una richiesta d'acquisto [180/settimana]
- Aggiornamento dello stato di un ordine [10/settimana]

Il secondo, invece, è coinvolto nelle operazioni di:

- Visualizzazione delle informazioni relative ad una Richiesta d'Acquisto [120/settimana];
- Inserimento di una Richiesta d'Acquisto [60/settimana].

4.1.3 Data di Consegna

Per ogni operazione, si prevedono gli accessi seguenti:

Visualizzazione degli articoli di una Richiesta d'Acquisto

		Presenza di attributo derivato		Assenza di attributo derivato	
<i>Concetto</i>	<i>Tipo</i>	<i>Accessi</i>	<i>Tipo di accesso</i>	<i>Accessi</i>	<i>Tipo di accesso</i>
Richiesta d'Acquisto	E	1	R	1	R
Include	R	5	R	5	R
Ordine	E	-	-	5	R

Aggiornamento dello stato di un ordine

		Presenza di attributo derivato		Assenza di attributo derivato	
<i>Concetto</i>	<i>Tipo</i>	<i>Accessi</i>	<i>Tipo di accesso</i>	<i>Accessi</i>	<i>Tipo di accesso</i>
Ordine	E	1	W	1	W
Include	R	60	W	-	-

Considerando la tavola dei volumi riportata precedentemente, si osserva quanto segue:

- L'operazione di *Visualizzazione degli articoli di una Richiesta d'Acquisto* considera:
 - 0 scritture e 6 letture in caso di presenza dell'attributo derivato
 - 0 scritture e 11 letture in caso di assenza dell'attributo derivato
- L'operazione di *Aggiornamento dello stato di un Ordine* considera:
 - 61 scritture e 0 letture in caso di presenza dell'attributo derivato
 - 1 scrittura e 0 letture in caso di assenza dell'attributo derivato

Applicando alle scritture un peso doppio rispetto a quello delle letture e considerando la frequenza delle operazioni sopracitate si osservano i costi di seguito descritti:

Nel caso di **presenza** dell'attributo derivato:

$$180 \cdot (0 \cdot 2 + 6 \cdot 1) + 10 \cdot (61 \cdot 2 + 0 \cdot 1) = 1080 + 1220 = 2300$$

Nel caso di **assenza** dell'attributo derivato:

$$180 \cdot (0 \cdot 2 + 11 \cdot 1) + 10 \cdot (1 \cdot 2 + 0 \cdot 1) = 1980 + 20 = 2000$$

Sulla base dei risultati ottenuti si sceglie, quindi, di non mantenere l'attributo derivato.

4.1.4 Numero Articoli

Per ogni operazione, si prevedono gli accessi seguenti:

Visualizzazione delle informazioni relative ad una Richiesta d'Acquisto

<i>Concetto</i>	<i>Tipo</i>	Presenza di attributo derivato		Assenza di attributo derivato	
		<i>Accessi</i>	<i>Tipo di accesso</i>	<i>Accessi</i>	<i>Tipo di accesso</i>
Richiesta d'Acquisto	E	1	R	1	R
Include	R	-	-	5	R

Inserimento di una Richiesta d'Acquisto

<i>Concetto</i>	<i>Tipo</i>	Presenza di attributo derivato		Assenza di attributo derivato	
		<i>Accessi</i>	<i>Tipo di accesso</i>	<i>Accessi</i>	<i>Tipo di accesso</i>
Richiesta d'Acquisto	E	1	R	1	R
Richiesta d'Acquisto	E	2	W	1	W
Include	R	5	R	-	-
Include	R	5	W	5	W

Considerando la tavola dei volumi riportata precedentemente, si osserva quanto segue:

- L'operazione di *Visualizzazione delle informazioni relative ad una Richiesta d'Acquisto* considera:
 - 0 scritture ed 1 lettura in caso di presenza dell'attributo derivato
 - 0 scritture e 6 letture in caso di assenza dell'attributo derivato
- L'operazione di *Inserimento di una Richiesta d'Acquisto* considera:
 - 7 scritture e 6 letture in caso di presenza dell'attributo derivato
 - 6 scritture e 1 lettura in caso di assenza dell'attributo derivato

Applicando alle scritture un peso doppio rispetto a quello delle letture e considerando la frequenza delle operazioni sopracitate si osservano i costi di seguito descritti:

Nel caso di **presenza** dell'attributo derivato:

$$120 \cdot (0 \cdot 2 + 1 \cdot 1) + 60 \cdot (7 \cdot 2 + 6 \cdot 1) = 120 + 1200 = 1320$$

Nel caso di **assenza** dell'attributo derivato:

$$120 \cdot (0 \cdot 2 + 6 \cdot 1) + 60 \cdot (6 \cdot 2 + 1 \cdot 1) = 720 + 780 = 1500$$

Sulla base dei risultati ottenuti si sceglie, quindi, di mantenere l'attributo derivato, procedendone alla reifica ad attributo nell'entità *Richiesta d'Acquisto*.

4.2 Eliminazione delle generalizzazioni

Non essendovi relazioni di generalizzazione nel diagramma concettuale proposto al punto 3.1, non è stato necessario apportare modifiche rivolte alla loro eliminazione.

4.3 Partizionamento ed accorpamento di entità e associazioni

4.3.1 Reifica di relazioni binarie

Il diagramma presenta una relazione binaria **Fornisce** che coinvolge le entità **Articolo** e **Fornitore**, che hanno entrambe una partecipazione di tipo (1, N). In particolare, per ogni coppia Articolo-Fornitore si osserva la presenza di una serie di attributi quali prezzo unitario, sconto, quantità minima ordinabile e codice articolo per il fornitore. Si sceglie, pertanto, di reificare la relazione ad un'omonima entità contenente gli attributi citati.

4.3.2 Reifica delle relazioni ternarie

Il diagramma ER presenta una relazione ternaria **Include** che coinvolge le entità **Richiesta d'Acquisto**, **Articolo** e **Ordine**. In particolare, la partecipazione delle entità Richiesta d'Acquisto e Articolo è di tipo (1, N), mentre quella dell'entità Ordine è (0, N): questo perché una richiesta non può essere vuota e un articolo può essere contenuto in una o più richieste, mentre un articolo appartenente ad una richiesta può non essere necessariamente soddisfatto da un ordine.

Al fine di eliminare la relazione ternaria, si sceglie di reificarla ad entità, in relazione con **Richiesta d'Acquisto**, **Articolo** ed **Ordine** ed avente come attributi quelli precedentemente individuati rispetto alla relazione.

4.3.3 Valutazione degli attributi composti

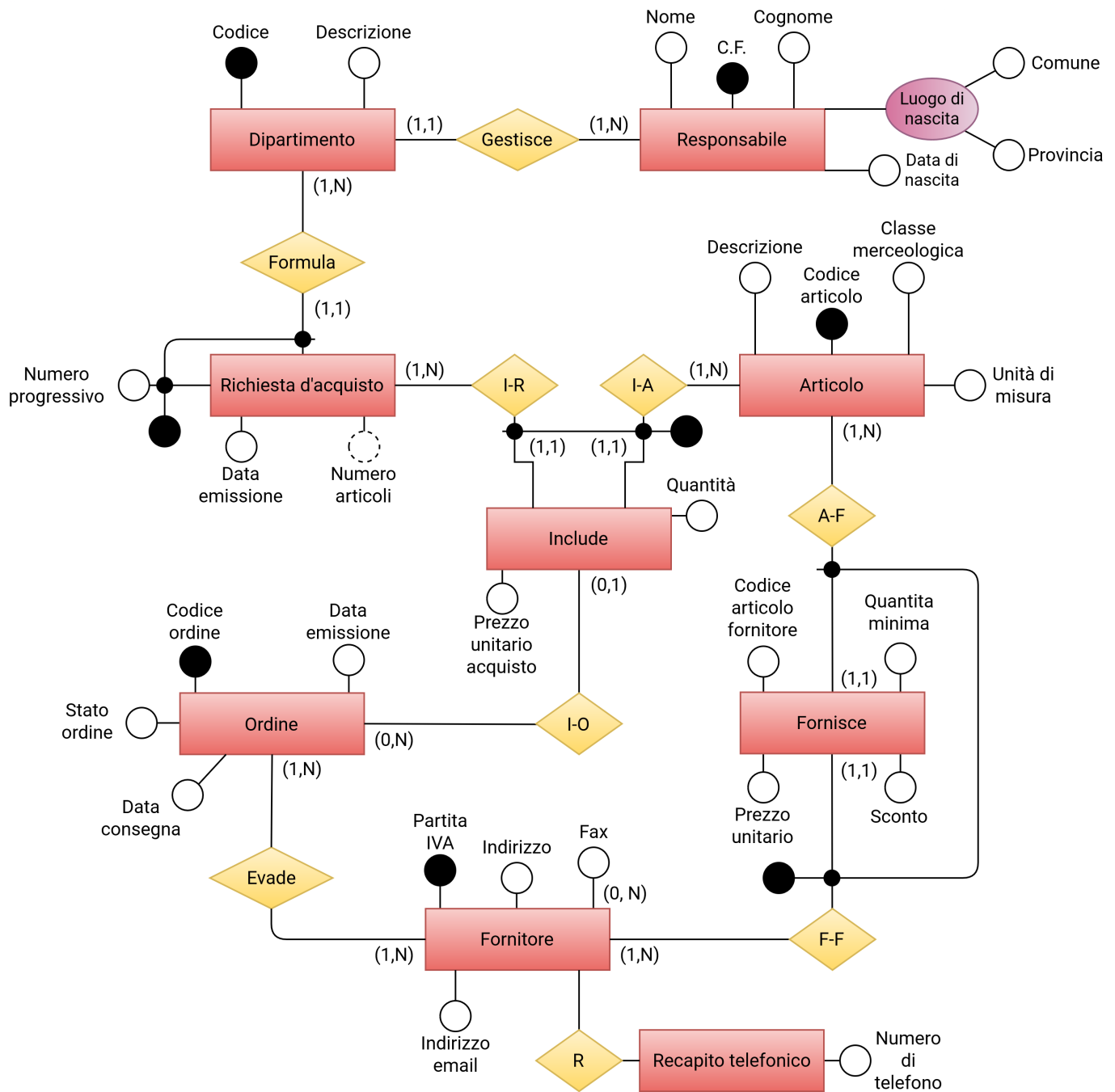
L'unico attributo composto presente nel diagramma è *Luogo di Nascita* in riferimento all'entità **Responsabile**. In particolare, l'attributo comprende i riferimenti relativi al Comune e alla Provincia di nascita. Vista la scarsità di interrogazioni in merito a dati anagrafici dei responsabili, si sceglie di mantenere l'attributo *Luogo di Nascita* rispetto alla separazione degli attributi *Comune* e *Provincia*. Si prevede, quindi, la presenza di un unico attributo contenente entrambe le informazioni.

4.3.4 Eliminazione di attributi multivalore

Il diagramma presenta un attributo multivalore *Recapiti Telefonici* in riferimento all'entità **Fornitore**. Questo, infatti, può avere uno o più contatti di riferimento. L'attributo multivalore viene, conseguentemente, reificato ad entità.

4.3.5 Ristrutturazione del diagramma ER

Sulla base delle analisi e osservazioni effettuate, si provvede alla ristrutturazione del diagramma proposto al punto 3.1. Ne consegue la seguente rappresentazione:



4.4 Scelta degli identificatori primari

Non essendovi entità che presentano più identificatori primari candidati, non si attuano decisioni aggiuntive e si sceglie di utilizzare le chiavi proposte dal diagramma.

4.5 Traduzione verso il modello logico-relazionale

Partendo dal diagramma ER ristrutturato, è stato prodotto il corrispondente schema relazionale, le cui traduzioni vengono di seguito suddivise in quattro categorie:

- Entità
- Relazioni molti a molti
- Relazioni uno a molti
- Relazioni uno a uno

Concetto	Cardinalità	Nome
Entità	-	Responsabile
Entità	-	Dipartimento
Entità	-	Richiesta d'Acquisto
Entità	-	Include
Entità	-	Articolo
Entità	-	Ordine
Entità	-	Fornisce
Entità	-	Fornitore
Entità	-	Recapito Telefonico
Relazione	Uno a molti	Gestisce
Relazione	Uno a molti	Formula
Relazione	Uno a molti	I-R
Relazione	Uno a molti	I-A
Relazione	Uno a molti	I-O
Relazione	Uno a molti	A-F
Relazione	Uno a molti	F-F
Relazione	Uno a molti	Evade
Relazione	Uno a molti	R

4.5.1 Traduzione di Entità

- **Responsabile**(CodiceFiscale, Nome, Cognome, DataNascita, LuogoNascita)
 - NotNull: Nome, Cognome, DataNascita, LuogoNascita
- **Dipartimento**(Codice, Descrizione)
- **RichiestaAcquisto**(Numero, Dipartimento, DataEmissione, NumeroArticoli)
 - NotNull: DataEmissione, Dipartimento, NumeroArticoli
 - Chiave Esterna: Dipartimento si riferisce alla chiave primaria dell'entità Dipartimento
- **Include**(NumeroRichiesta, Articolo, Dipartimento, Quantità, PrezzoUnitario)
 - NotNull: Quantità, PrezzoUnitario, NumeroRichiesta, Dipartimento, Articolo
 - Chiave Esterna: NumeroRichiesta e Dipartimento si riferiscono alla chiave primaria dell'entità RichiestaAcquisto, Articolo si riferisce alla chiave primaria dell'entità Articolo
- **Articolo**(Codice, Descrizione, Classe, UnitàDiMisura)
 - NotNull: Descrizione, Classe, UnitàDiMisura
- **Ordine**(Codice, Stato, DataEmissione, DataConsegna)
 - NotNull: Stato, DataEmissione

- **Fornisce**(Fornitore, Articolo, Sconto, PrezzoUnitario, QuantitàMinima, CodBar)
 - NotNull: PrezzoUnitario, QuantitàMinima, CodBar, Fornitore, Articolo
 - Chiave Esterna: Fornitore si riferisce alla chiave primaria dell'entità Fornitore, Articolo si riferisce alla chiave primaria dell'entità Articolo
- **Fornitore**(PartitaIVA, Indirizzo, Email, FAX)
 - NotNull: Indirizzo, Email
- **RecapitoTelefonico**(NumeroTelefono)

4.5.2 Traduzione di Relazioni Uno a Molti

I vincoli espressi di seguito costituiscono un'integrazione rispetto a quelli introdotti precedentemente.

- **Gestisce**
 - Modifica: Dipartimento(Codice, Descrizione, *Responsabile*)
 - NotNull: Responsabile
 - Chiave Esterna: Responsabile si riferisce alla chiave primaria dell'entità Responsabile
- **Formula**
 - Codificata precedentemente in quanto Richiesta d'Acquisto è un'entità debole
- **I-R e I-A**
 - Codificate precedentemente in quanto Include è un'entità debole)
- **I-O**
 - Modifica: Include(NumeroRichiesta, Articolo, Dipartimento, Ordine, Quantità, PrezzoUnitario)
 - NotNull: Non vengono introdotti vincoli aggiuntivi rispetto a quelli già individuati
 - Chiave Esterna: Ordine si riferisce alla chiave primaria dell'entità Ordine
- **A-F e F-F**
 - Codificate precedentemente in quanto Fornisce è un'entità debole)
- **Evade**
 - Modifica: Ordine(Codice, Stato, DataEmissione, DataConsegna, *Fornitore*)
 - NotNull: Fornitore
 - Chiave Esterna: Fornitore si riferisce alla chiave primaria dell'entità Fornitore
- **R**
 - Modifica: RecapitoTelefonico(NumeroTelefono, *Fornitore*)
 - NotNull: Fornitore
 - Chiave Esterna: Fornitore si riferisce alla chiave primaria dell'entità Fornitore

4.5.3 Traduzione di relazioni molti a molti e uno a uno

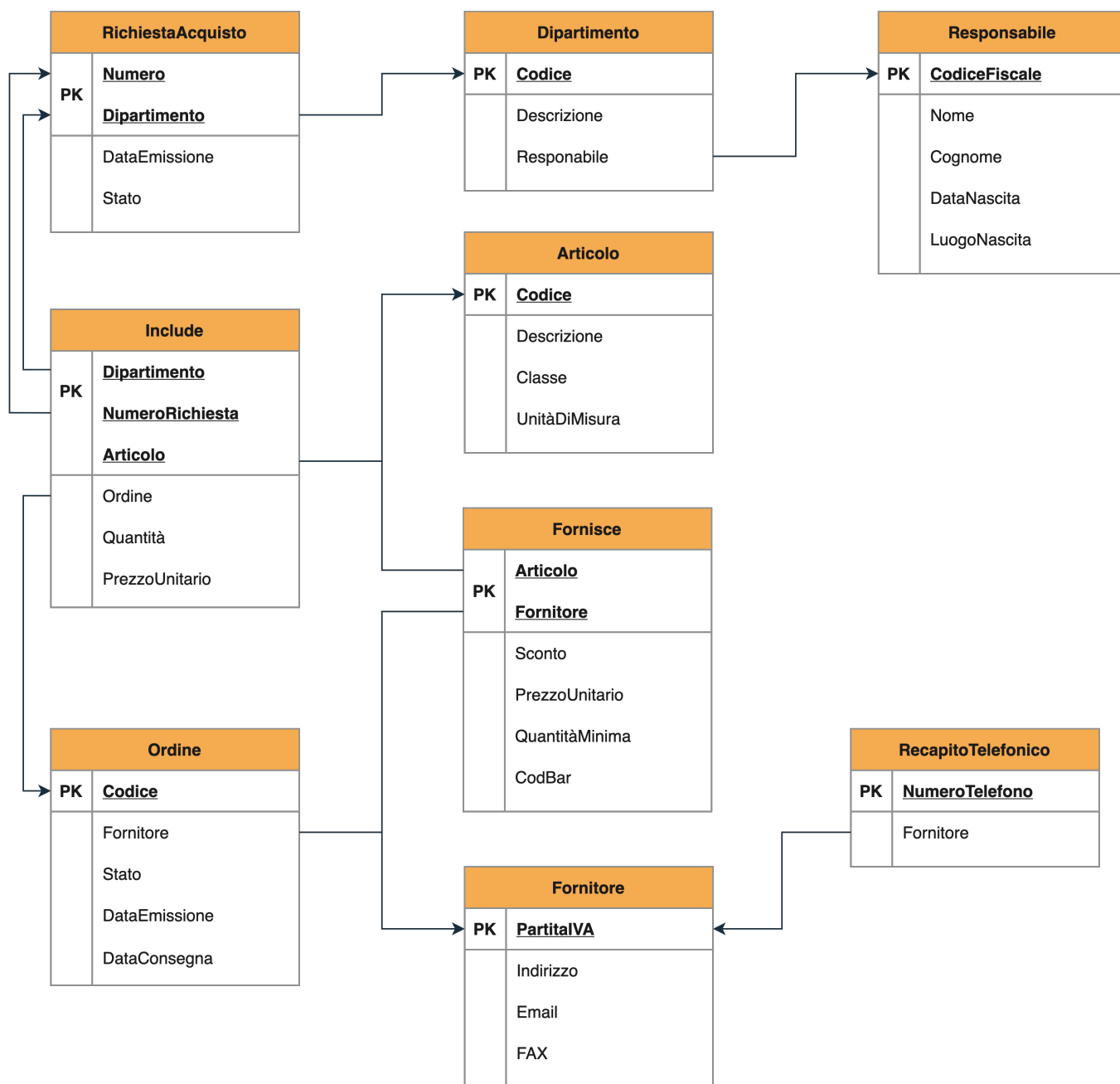
Il diagramma ER non presenta relazioni di tipo *molti a molti* e di tipo *uno a uno*. Di conseguenza non vi è necessità di codificare relazioni di questo tipo.

4.5.4 Osservazioni

Si osserva come non sia possibile garantire il rispetto del Vincolo di Integrità espresso al punto 3.2.1. Sarà, di conseguenza, necessario individuare appositi strumenti al fine di garantirne il mantenimento.

4.6 Modello Relazionale

Sulla base delle osservazioni effettuate, si provvede alla rappresentazione del diagramma relazionale:



5 Progettazione Fisica

5.1 Osservazioni sugli indici

Al fine di introdurre un miglioramento delle prestazioni, si valuta l'inserimento di ulteriori indici confrontando la variazione delle prestazioni sia in operazioni di **ricerca** che in operazioni di **modifica**. L'indicizzazione permette, infatti, un tempo di lookup inferiore durante query di selezione ma può causare l'aumento dei tempi di esecuzione delle query di modifica e inserimento sulla stessa tabella. Si rende, pertanto, necessario un confronto atto a stabilire le variazioni che i tempi di esecuzione subiscono in entrambi i casi.

A tal fine, è stato utilizzato il comando `EXPLAIN ANALYZE [statement]`, che permette di ottenere informazioni sull'**execution plan** e sui tempi di esecuzione richiesti da una query. È stato, inoltre, impostato ad **OFF** l'attributo `enable seqscan` al fine di scoraggiare il query planner all'utilizzo di scan sequenziali che invaliderebbero i confronti fra operazioni su tabelle in assenza e presenza di indici.

Si tiene, inoltre, presente il fatto che ogni tabella viene automaticamente indicizzata dal DBMS sulla sua chiave primaria.

Gli indici presi in considerazione sono i seguenti:

- Indicizzazione sugli attributi **Dipartimento** e **NumeroRichiesta** dell'entità *Include*
- Indicizzazione sull'attributo **Ordine** dell'entità *Include*
- Indicizzazione sull'attributo **DataRichiesta** dell'entità *RichiestaAcquisto*

Nel primo caso, è stato osservato come l'indicizzazione di chiavi primarie composite in PostgreSQL avvenga anche su sottoinsiemi delle stesse. Pertanto, considerata l'appartenenza di Dipartimento e NumeroRichiesta alla chiave primaria di RichiestaAcquisto, non risulterebbe conveniente l'aggiunta di un ulteriore indice sui due soli attributi. Il DBMS sfrutterebbe, in ogni caso, l'indicizzazione della chiave primaria. Si sceglie, pertanto, di non implementare tale indice all'interno della base di dati.

Nel secondo e terzo caso, invece, si sceglie di procedere al confronto in presenza e assenza degli indici. L'indicizzazione dell'entità *Include* sull'attributo **Ordine** permetterebbe, infatti, una più efficiente ricerca degli articoli contenuti in un determinato Ordine, mentre quella dell'entità *RichiestaAcquisto* sull'attributo **DataEmissione** permetterebbe una più veloce ricerca delle Richieste d'Acquisto effettuate in un determinato intervallo di tempo, utile durante la computazione di statistiche e metriche mensili, trimestrali e annuali da parte dell'ente pubblico.

I test sono stati condotti sui dati di Mockup (la cui produzione viene descritta successivamente), realizzati nel rispetto dei volumi descritti al punto 4.1.2 al fine di poter condurre operazioni di test e di analisi sulla base di dati.

L'ottenimento dei tempi di planning ed esecuzione e la successiva produzione dei rispettivi grafici è stato, invece, delegato allo script `IndexEval.R`, che utilizza la libreria [RPostgreSQL](#) ed è localizzato all'interno della directory `R`.

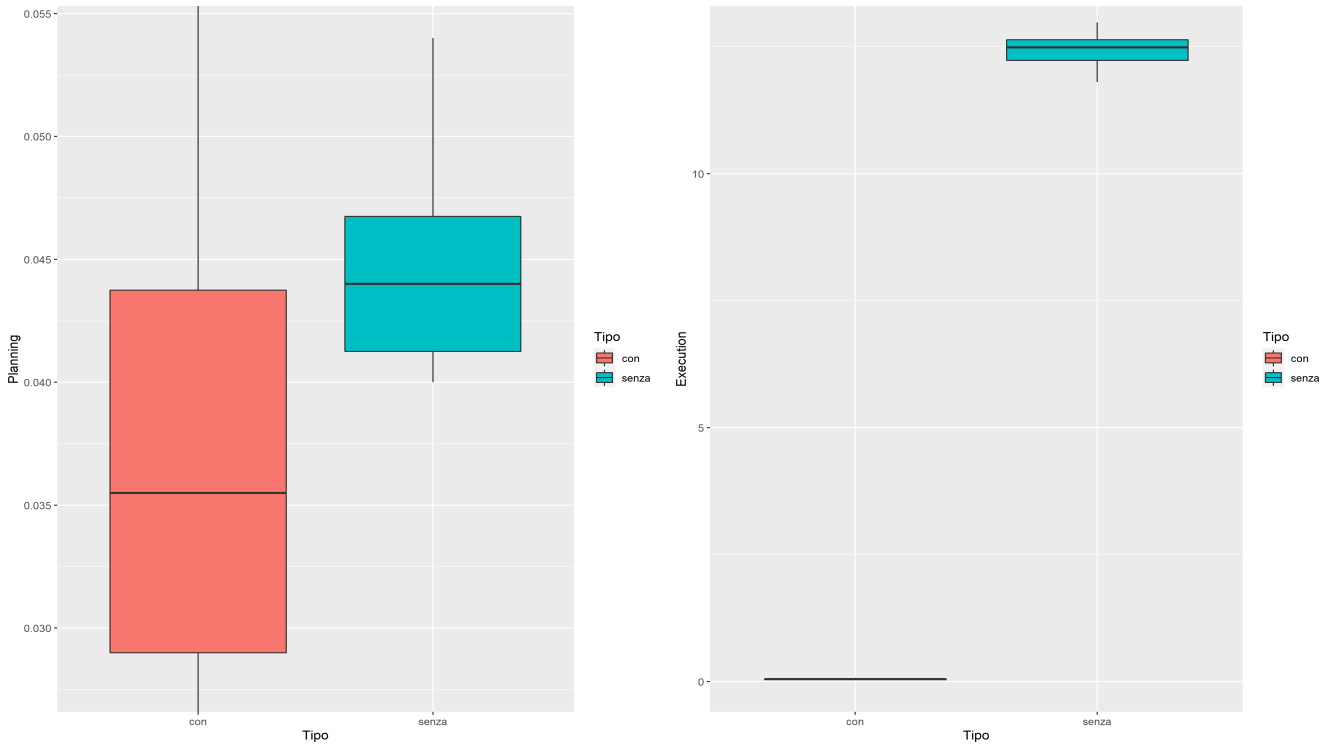
5.1.1 Indicizzazione di Include su Ordine in operazioni di ricerca

Assenza dell'Indice		Presenza dell'Indice	
<i>Planning</i>	<i>Execution</i>	<i>Planning</i>	<i>Execution</i>
0.041	12.508	0.030	0.039
0.042	12.135	0.025	0.035
0.041	17.798	0.029	0.039
0.054	24.227	0.024	0.036
0.053	17.478	0.024	0.038
0.043	12.222	0.044	0.036
0.046	12.481	0.039	0.054
0.047	12.406	0.043	0.056
0.044	12.968	0.055	0.073
0.041	12.377	0.039	0.053
0.041	12.271	0.028	0.050
0.042	12.405	0.027	0.037
0.044	12.177	0.051	0.063
0.045	12.798	0.035	0.043
0.042	12.493	0.034	0.055
0.041	12.146	0.042	0.068
0.047	12.522	0.047	0.057
0.040	12.490	0.043	0.055
0.044	12.245	0.044	0.048
0.042	12.541	0.030	0.044
0.053	12.546	0.036	0.042
0.046	13.329	0.044	0.070
0.041	12.556	0.048	0.064
0.041	12.462	0.032	0.042
0.046	12.637	0.025	0.037
0.044	12.509	0.034	0.057
0.045	12.396	0.066	0.088
0.042	12.571	0.031	0.043
0.044	12.308	0.051	0.044
0.063	12.423	0.039	0.051
0.041	12.105	0.055	0.066
0.045	12.980	0.054	0.068
0.060	12.125	0.082	0.063
0.041	12.290	0.026	0.038
0.051	12.699	0.041	0.052
0.053	12.491	0.028	0.038
0.047	12.550	0.039	0.052
0.048	11.804	0.033	0.040
0.044	12.663	0.025	0.036
0.044	12.212	0.056	0.040
0.044	12.006	0.031	0.044
0.041	12.740	0.034	0.046
0.042	12.276	0.043	0.048
0.041	12.113	0.026	0.039
0.054	16.108	0.029	0.044
0.041	12.631	0.024	0.050
0.044	12.647	0.032	0.044
0.055	12.048	0.042	0.053
0.042	12.021	0.037	0.042
0.043	11.963	0.023	0.074

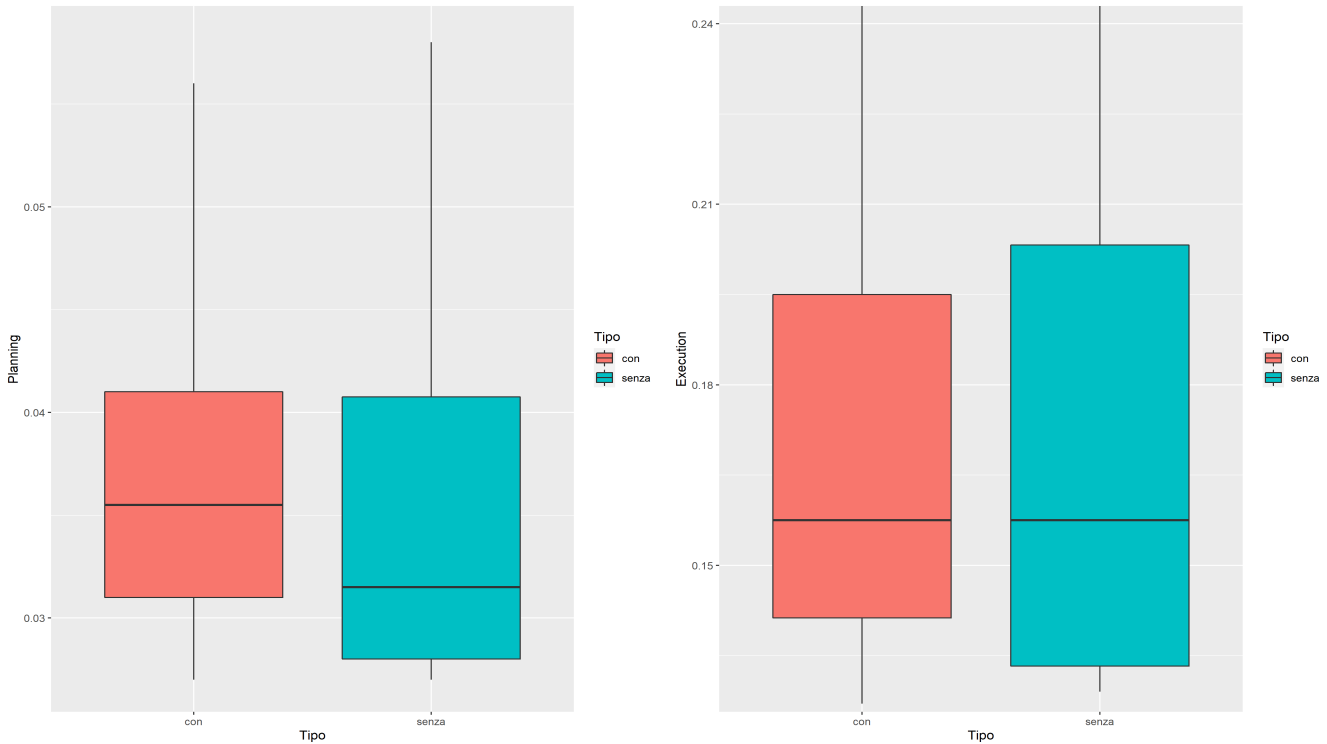
5.1.2 Indicizzazione di Include su Ordine in operazioni di inserimento

Assenza dell'Indice		Presenza dell'Indice	
<i>Planning</i>	<i>Execution</i>	<i>Planning</i>	<i>Execution</i>
0.072	0.677	0.037	0.216
0.028	0.424	0.052	0.145
0.033	0.393	0.034	0.163
0.037	0.158	0.037	0.149
0.027	0.133	0.033	0.133
0.064	0.233	0.031	0.148
0.027	0.131	0.028	0.165
0.028	0.131	0.031	0.148
0.027	0.130	0.031	0.149
0.027	0.129	0.158	0.261
0.029	0.135	0.031	0.183
0.029	0.132	0.037	0.152
0.028	0.130	0.030	0.145
0.027	0.137	0.030	0.139
0.034	0.183	0.029	0.134
0.027	0.135	0.041	0.165
0.032	0.159	0.049	0.197
0.027	0.130	0.062	0.309
0.038	0.165	0.043	0.201
0.028	0.133	0.039	0.175
0.056	0.221	0.041	0.226
0.027	0.184	0.041	0.210
0.033	0.135	0.034	0.230
0.027	0.129	0.041	0.268
0.041	0.230	0.036	0.259
0.028	0.133	0.031	0.140
0.049	0.182	0.029	0.133
0.045	0.212	0.033	0.133
0.031	0.139	0.045	0.218
0.044	0.189	0.030	0.138
0.036	0.169	0.029	0.132
0.030	0.208	0.038	0.189
0.038	0.150	0.064	0.252
0.031	0.149	0.029	0.133
0.029	0.134	0.035	0.161
0.032	0.177	0.038	0.165
0.027	0.136	0.029	0.136
0.040	0.167	0.042	0.175
0.083	0.237	0.039	0.178
0.037	0.159	0.036	0.152
0.068	0.243	0.028	0.127
0.063	0.229	0.027	0.127
0.060	0.180	0.032	0.146
0.030	0.142	0.052	0.168
0.037	0.157	0.055	0.154
0.028	0.135	0.056	0.152
0.027	0.131	0.032	0.128
0.027	0.131	0.034	0.153
0.058	0.219	0.034	0.215
0.073	0.261	0.038	0.174

5.1.2.1 Osservazioni Sulla base dei dati ottenuti sono stati prodotti i seguenti grafici:



Variazione di Planning ed Execution time per operazioni di selezione



Variazione di Planning ed Execution time per operazioni di modifica

Le query di selezione e modifica utilizzate sono le seguenti:

```
-- Selezione
EXPLAIN ANALYSE
  SELECT *
  FROM Include
  WHERE Ordine=5;

-- Modifica
EXPLAIN ANALYSE
  UPDATE Include
  SET Ordine=NULL
  WHERE
    Dipartimento='WLIQJC' AND
    NumeroRichiesta=79 AND
    Articolo=102;
```

Si osserva quanto segue:

- Nel caso di **query di selezione** i tempi di esecuzione migliorano notevolmente in presenza di un indice
- Nel caso di **query di modifica** la presenza dell'indice non causa notevoli variazioni nei tempi di esecuzione

Si sceglie, pertanto, di **mantenere l'indice** all'interno della base di dati.

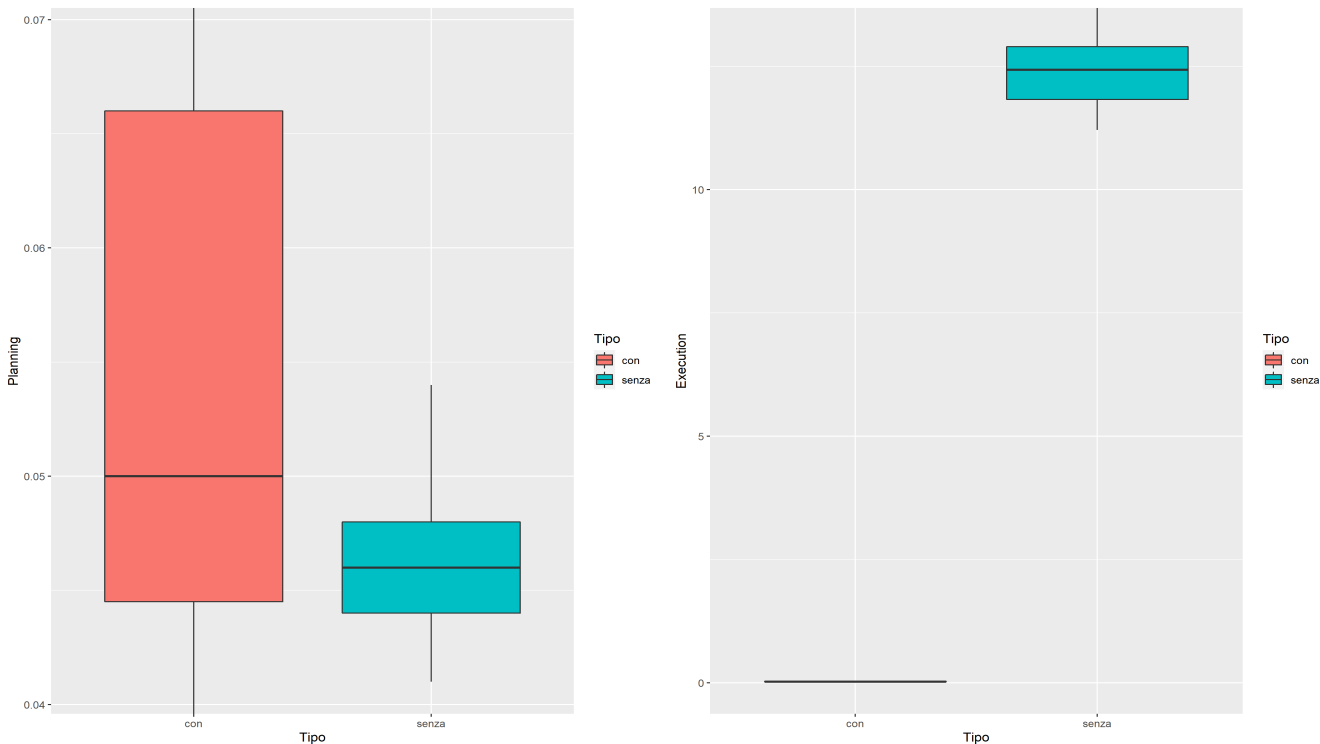
5.1.3 Indicizzazione di DataEmissione su RichiestaAcquisto in operazioni di ricerca

Assenza dell'Indice		Presenza dell'Indice	
<i>Planning</i>	<i>Execution</i>	<i>Planning</i>	<i>Execution</i>
0.043	12.897	0.068	0.042
0.056	12.058	0.050	0.029
0.044	12.548	0.044	0.025
0.046	13.336	0.038	0.024
0.048	13.105	0.049	0.031
0.044	11.490	0.058	0.024
0.047	14.510	0.053	0.026
0.062	19.816	0.068	0.041
0.047	12.892	0.047	0.024
0.046	12.581	0.078	0.046
0.046	12.343	0.049	0.025
0.048	12.893	0.038	0.023
0.048	11.667	0.063	0.025
0.041	12.688	0.046	0.028
0.044	12.468	0.048	0.029
0.042	11.733	0.037	0.023
0.044	11.276	0.042	0.024
0.042	11.326	0.039	0.024
0.042	11.205	0.071	0.026
0.042	11.536	0.066	0.038
0.046	11.529	0.037	0.023
0.044	11.602	0.050	0.032
0.047	12.184	0.051	0.026
0.047	11.938	0.066	0.057
0.044	11.719	0.049	0.033
0.044	11.806	0.070	0.041
0.066	13.281	0.075	0.033
0.046	12.417	0.048	0.029
0.054	12.981	0.047	0.032
0.044	11.667	0.053	0.026
0.075	12.504	0.190	0.025
0.043	12.676	0.057	0.024
0.044	12.154	0.039	0.036
0.047	14.900	0.041	0.025
0.051	16.153	0.058	0.024
0.053	13.379	0.041	0.025
0.058	13.122	0.070	0.025
0.043	12.436	0.069	0.032
0.044	11.883	0.039	0.024
0.044	11.921	0.070	0.025
0.067	13.741	0.061	0.039
0.043	12.835	0.078	0.047
0.048	12.492	0.061	0.036
0.054	12.083	0.070	0.045
0.044	11.749	0.050	0.027
0.051	11.965	0.038	0.023
0.044	12.670	0.050	0.025
0.046	12.027	0.057	0.033
0.048	11.909	0.049	0.028
0.055	13.019	0.037	0.022

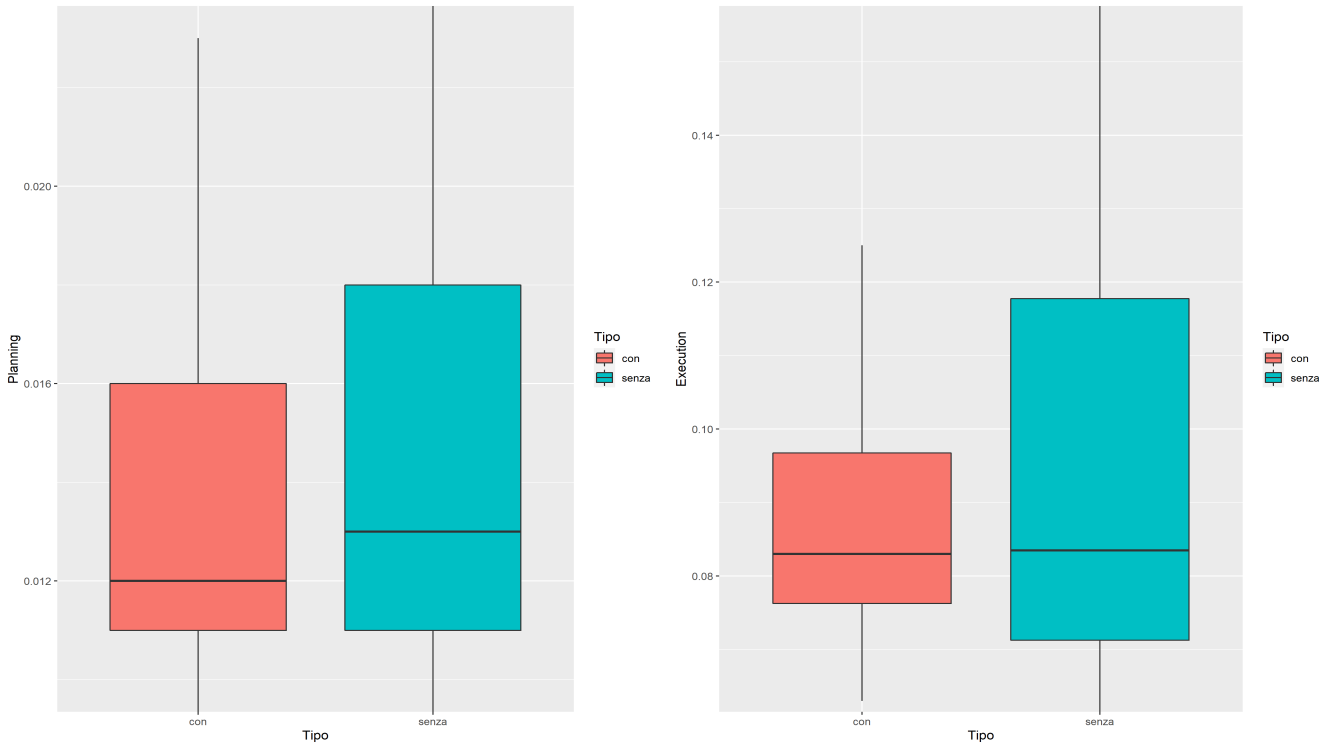
5.1.4 Indicizzazione di DataEmissione su RichiestaAcquisto in operazioni di modifica

Assenza dell'Indice		Presenza dell'Indice	
<i>Planning</i>	<i>Execution</i>	<i>Planning</i>	<i>Execution</i>
0.016	0.192	0.019	0.094
0.011	0.171	0.009	0.064
0.010	0.126	0.012	0.076
0.014	0.113	0.010	0.066
0.012	0.178	0.022	0.122
0.014	0.153	0.015	0.097
0.014	0.089	0.017	0.092
0.013	0.079	0.017	0.085
0.013	0.074	0.011	0.067
0.013	0.072	0.011	0.072
0.011	0.064	0.009	0.063
0.012	0.071	0.013	0.107
0.015	0.083	0.011	0.085
0.009	0.062	0.009	0.066
0.013	0.095	0.010	0.066
0.010	0.069	0.010	0.066
0.020	0.098	0.023	0.129
0.012	0.072	0.009	0.106
0.009	0.064	0.012	0.080
0.009	0.063	0.012	0.084
0.022	0.127	0.011	0.077
0.011	0.066	0.011	0.079
0.018	0.080	0.010	0.064
0.010	0.066	0.021	0.181
0.013	0.081	0.016	0.119
0.015	0.123	0.023	0.096
0.021	0.111	0.024	0.208
0.010	0.119	0.012	0.084
0.010	0.065	0.011	0.078
0.011	0.077	0.012	0.087
0.013	0.067	0.012	0.082
0.009	0.113	0.012	0.080
0.026	0.155	0.013	0.080
0.018	0.111	0.012	0.082
0.018	0.118	0.011	0.077
0.035	0.315	0.012	0.116
0.026	0.218	0.012	0.079
0.023	0.101	0.012	0.080
0.014	0.075	0.016	0.105
0.021	0.194	0.010	0.069
0.016	0.075	0.023	0.131
0.016	0.084	0.031	0.237
0.018	0.107	0.026	0.125
0.018	0.084	0.010	0.087
0.021	0.117	0.015	0.090
0.016	0.081	0.012	0.071
0.013	0.069	0.012	0.082
0.009	0.060	0.010	0.074
0.012	0.078	0.026	0.084
0.010	0.062	0.014	0.089

5.1.4.1 Osservazioni Sulla base dei dati ottenuti sono stati prodotti i seguenti grafici:



Variazione di Planning ed Execution time per operazioni di selezione



Variazione di Planning ed Execution time per operazioni di modifica

Le query di selezione e modifica utilizzate sono le seguenti:

```
-- Selezione
EXPLAIN ANALYSE
  SELECT *
  FROM RichiestaAcquisto
  WHERE DataEmissione BETWEEN '2020-10-01' AND '2020-11-01'

-- Modifica
EXPLAIN ANALYSE
  INSERT INTO RichiestaAcquisto(Dipartimento)
  VALUES ('ZXTSNW')
```

Si osserva quanto segue:

- Nel caso di **query di selezione** i tempi di esecuzione subiscono un notevole miglioramento in presenza dell'indice
- Nel caso di **query di modifica** i tempi di esecuzione non subiscono variazioni significative, anche se si osserva una maggiore variabilità nel caso di assenza dell'indice.

Sulla base dei risultati ottenuti si sceglie, quindi, il **mantenimento dell'indice**.

6 Implementazione

6.1 Containerizzazione del DBMS

Al fine di agevolare il processo di implementazione e deployment, si è scelto di utilizzare un container docker basato sull'immagine [postgres](#). Di conseguenza, è stato descritto il seguente `docker-compose.yml`:

```
version: "3.9"
services:
  db:
    image: postgres
    container_name: db
    ports:
      - "15000:5432"
    volumes:
      - ./db:/var/lib/postgresql/data
    environment:
      POSTGRES_PASSWORD: bdd2021
```

È, quindi, possibile accedere al DBMS tramite le seguenti credenziali:

Parametro	Valore
Utente	postgres
Password	bdd2021
Indirizzo	localhost
Porta	15000

Il contenuto del DBMS viene serializzato all'interno della directory `psqlOnDocker/db`.

6.2 SQL

6.2.1 Definizione dei tipi enum

Sulla base di quanto individuato nel corso dell'analisi, sono stati definiti i tipi di dato atti a descrivere le possibili classi merceologiche di un articolo, le unità di misura e gli stati di un ordine.

```
create type classe_merceologica as enum (
  'cancelleria',
  'libri',
  'elettronica',
  'informatica',
  'pulizia',
  'mobilia'
);

create type unita_misura as enum (
  'cad',
  'kg',
  'm',
  'l'
);

create type stato_ordine as enum (
  'emesso',
  'spedito',
  'consegnato',
  'annullato'
);
```

6.2.2 Creazione delle tabelle

Di seguito, sono state definite le tabelle (con rispettivi vincoli di chiave primaria e chiave esterna) sulla base di quando descritto dal diagramma relazionale presentato al punto 4.6.

```
create table Responsabile
(
    CodiceFiscale char(16) primary key,
    Nome text not null,
    Cognome text not null,
    DataNascita date not null,
    LuogoNascita text not null
);

create table Dipartimento
(
    Codice char(6) primary key,
    Descrizione text not null,
    Responsabile char(16) not null
        references Responsabile
        on update cascade
        on delete restrict
);

create table RichiestaAcquisto
(
    Numero integer,
    Dipartimento char(6) not null
        references Dipartimento
        on update cascade
        on delete restrict,
    DataEmissione date not null default current_date,
    NumeroArticoli integer not null default 0,
    primary key (Numero, Dipartimento)
);

create table Articolo
(
    Codice serial primary key,
    Descrizione text not null,
    Classe classe_merceologica not null,
    UnitàDiMisura unità_misura not null
);

create table Fornitore
(
    PartitaIVA char(13) primary key,
    Indirizzo text not null,
    Email varchar(50) not null,
    FAX varchar(15)
);
```

```

create table RecapitoTelefonico
(
    NumeroTelefono varchar(15) primary key,
    Fornitore char(13) not null
        references Fornitore
        on update cascade
        on delete cascade
);

create table Fornisce
(
    Articolo integer
        references Articolo
        on update cascade
        on delete cascade,
    Fornitore char(13)
        references Fornitore
        on update cascade
        on delete cascade,
    Sconto numeric not null default 0,
    PrezzoUnitario numeric not null
        check (PrezzoUnitario > 0),
    QuantitaMinima integer not null default 1
        check (QuantitaMinima >= 1),
    CodBar varchar(20) not null,
    primary key (Articolo, Fornitore)
);

create table Ordine
(
    Codice serial primary key,
    Fornitore char(13) not null
        references Fornitore
        on update cascade
        on delete restrict,
    Stato stato_ordine not null default 'emesso',
    DataEmissione date not null default current_date,
    DataConsegna date default null
);

create table Include
(
    Dipartimento char(6),
    NumeroRichiesta integer,
    Articolo integer
        references Articolo
        on update cascade
        on delete restrict,
    Ordine integer default null
        references Ordine
        on update cascade
        on delete set null,

```

```

Quantita numeric not null
    check (Quantita > 0),
PrezzoUnitario numeric(7, 2) default null,
primary key (Dipartimento, NumeroRichiesta, Articolo),
foreign key (Dipartimento, NumeroRichiesta)
    references RichiestaAcquisto (Dipartimento, Numero)
    on update cascade
    on delete restrict
);

```

È stata, inoltre, implementata la tabella **ProssimoCodiceRichiesta**, che permette di mantenere in memoria il codice di una nuova eventuale Richiesta d'Acquisto per ognuno dei dipartimenti presenti. Ad esempio:

Dipartimento	ProssimoNumero
ZXTSNW	10
WPIUQD	3
...	...

L'aggiornamento dei campi al suo interno è permesso dai trigger descritti al punto successivo.

```

create table ProssimoCodiceRichiesta
(
    Dipartimento char(6) primary key
        references Dipartimento
        on update cascade
        on delete cascade,
    ProssimoNumero integer default 1
);

```


6.2.3 Definizione dei trigger

Sono stati, inoltre, definiti i trigger necessari al mantenimento del vincolo aziendale descritto al punto 3.2.1, alla sincronizzazione degli attributi derivati e al mantenimento di informazioni coerenti e consistenti all'interno della base di dati.

6.2.3.1 Vincolo aziendale

```
create or replace function controlla_ordine_valido()
    returns trigger
    language plpgsql as
$$
declare
    n      integer;
    forn   character(13);
begin
    if new.Ordine IS NULL then
        return new;
    end if;

    SELECT Fornitore
    INTO forn
    FROM Ordine
    WHERE Codice = new.Ordine;

    SELECT COUNT(*)
    INTO n
    FROM Fornisce
    WHERE Fornisce.Articolo = new.Articolo
        AND forn = Fornisce.Fornitore;

    if n = 0 then
        raise notice 'Prodotto non valido per fornitore';
        return null;
    end if;
    return new;
end;
$$;

create trigger controlla_ordine_valido
    before insert or update
    on Include
    for each row
execute procedure controlla_ordine_valido();
```

6.2.3.2 Calcolo del prezzo unitario con sconto

```
create or replace function calcola_prezzo_finale()
  returns trigger
  language plpgsql as
$$
declare
  currentOrder    integer;
  currentSupplier varchar;
  price           numeric;
  discount        numeric;
  finalPrice      numeric;
begin
  if new.Ordine is not null then
    currentOrder = new.Ordine;

    SELECT Fornitore
    INTO currentSupplier
    FROM Ordine
    WHERE Codice = currentOrder;

    SELECT PrezzoUnitario, Sconto
    INTO price, discount
    FROM Fornisce
    WHERE Fornitore = currentSupplier
           AND Articolo = new.Articolo;

    finalPrice = price * (1 - discount / 100);
    new.PrezzoUnitario = finalPrice;
  end if;
  return new;
end;
$$;

create trigger calcola_prezzo_finale
  before insert or update of Ordine
  on Include
  for each row
execute procedure calcola_prezzo_finale();
```

6.2.3.3 Verifica della possibile rimozione di un Ordine

```
create or replace function rimuovi_ordine()
  returns trigger
  language plpgsql as
$$
begin
  if old.Stato = 'consegnato' or old.stato = 'spedito' then
    raise exception 'Non puoi rimuovere questo ordine!';
  elseif old.Stato = 'emesso' then
    old.Stato = 'annullato';
  end if;

  UPDATE Include
  SET Ordine=NULL
  WHERE Ordine=old.Codice;

  return old;
end;
$$;

create trigger rimuovi_ordine
  before delete on Ordine
  for each row
execute procedure rimuovi_ordine();
```

Un ordine può essere, infatti, rimosso solamente se si trova in stato **annullato**. Nel caso in cui l'ordine sia nello stato **emesso**, il trigger procede autonomamente alla modifica dello stato e alla successiva cancellazione. Questo è motivato dal fatto che la cancellazione di un ordine emesso non può provocare inconsistenze nella base di dati.

Nel caso in cui l'ordine si trovi in uno degli stati rimanenti, la procedura di cancellazione non viene consentita e viene delegata all'utente della base di dati la responsabilità relativa alla modifica dello stato dell'ordine al fine di consentirne la cancellazione.

6.2.3.4 Sincronizzazione dell'attributo derivato NumeroArticoli

```
-- Inserimento in Include
create or replace function numero_articoli_aumenta()
  returns trigger
  language plpgsql as
$$
declare
  n_art integer;
begin

  UPDATE RichiestaAcquisto
  SET NumeroArticoli = NumeroArticoli + new.Quantita
  WHERE Dipartimento=new.Dipartimento
      AND Numero=new.NumeroRichiesta;

  return new;
end;
$$;

create trigger numero_articoli_aumenta
  before insert
  on Include
  for each row
execute procedure numero_articoli_aumenta();

-- Rimozione da Include
create or replace function numero_articoli_riduci()
  returns trigger
  language plpgsql as
$$
declare
  n_art integer;
begin
  UPDATE RichiestaAcquisto
  SET NumeroArticoli = NumeroArticoli - old.Quantita
  WHERE Dipartimento=old.Dipartimento
      AND Numero=old.NumeroRichiesta;
  return old;
end;
$$;

create trigger numero_articoli_riduci
  before delete
  on Include
  for each row
execute procedure numero_articoli_riduci();
```

```

-- Aggiornamento in Include
create or replace function numero_articoli_aggiorna()
  returns trigger
  language plpgsql as
$$
declare
  n_art integer;
begin

  UPDATE RichiestaAcquisto
  SET NumeroArticoli = NumeroArticoli - old.Quantita
  WHERE Dipartimento=old.Dipartimento
     AND Numero=old.NumeroRichiesta;

  UPDATE RichiestaAcquisto
  SET NumeroArticoli = NumeroArticoli + new.Quantita
  WHERE Dipartimento=new.Dipartimento
     AND Numero=new.NumeroRichiesta;

  return new;
end;
$$;

create trigger numero_articoli_aggiorna
  after update
  on Include
  for each row
execute procedure numero_articoli_aggiorna();

```

6.2.3.5 Verifica del rispetto della quantità minima ordinabile

```
create or replace function controlla_quantita_minima()
  returns trigger
  language plpgsql as
$$
declare
  q      integer;
  forn   character(13);
begin
  if new.Ordine IS NULL then
    return new;
  end if;

  SELECT Fornitore
  INTO forn
  FROM Ordine
  WHERE Codice = new.Ordine;

  SELECT QuantitaMinima
  INTO q
  FROM Fornisce
  WHERE (Fornisce.Articolo = new.Artic
        AND (forn = Fornisce.Fornitore);

  if new.Quantita < q then
    raise notice 'La quantità minima ordinabile non è soddisfatta';
    return null;
  end if;
  return new;
end;
$$;

create trigger controlla_quantita_minima
  before insert or update of Ordine
  on Include
  for each row
execute procedure controlla_quantita_minima();
```

6.2.3.6 Inserimento di un nuovo dipartimento in ProssimoCodiceRichiesta

```
create or replace function nuova_entry_dipartimento()
  returns trigger
  language plpgsql as
$$
begin

    INSERT INTO ProssimoCodiceRichiesta(Dipartimento) VALUES (new.Codice);

    return new;
end;
$$;

create trigger nuova_entry_dipartimento
  after insert
  on Dipartimento
  for each row
execute procedure nuova_entry_dipartimento();
```

6.2.3.7 Aggiornamento di ProssimoCodiceRichiesta

```
create or replace function set_numero_richiesta()
  returns trigger
  language plpgsql as
$$
declare
  n integer;
begin

  SELECT ProssimoNumero
  INTO n
  FROM ProssimoCodiceRichiesta
  WHERE Dipartimento = new.Dipartimento;

  if n is null then
    raise notice 'Errore: dipartimento non valido';
    return null;
  else
    new.numero := n;

    UPDATE ProssimoCodiceRichiesta
    SET ProssimoNumero = n+1
    WHERE Dipartimento = new.Dipartimento;

    return new;
  end if;
end;
$$;

create trigger set_numero_richiesta
  before insert
  on RichiestaAcquisto
  for each row
execute procedure set_numero_richiesta();
```


6.2.4 Definizione degli indici

Sulla base di quanto convenuto in precedenza, si sceglie di includere ulteriori indici per le entità **Include** e **RichiestaAcquisto**.

```
create index on Include(Ordine);
```

```
create index on RichiestaAcquisto(DataEmissione);
```

L'implementazione descritta è contenuta interamente nel file `psqlOnDocker/create_db.sql`.

6.3 Produzione ed Inserimento dei dati di Mockup

Al fine di popolare il DBMS con dati realistici e coerenti con i volumi dichiarati al punto 4.1.2, è stato realizzato uno script Python (`psqlOnDocker/MockupDataGenerator/script.py`) che sfrutta la libreria [Faker](#).

Quest'ultimo genera, per ognuna delle tabelle presenti all'interno della base di dati, un omonimo file **sql** contenente le query di inserimento. Al fine di rendere i dati quanto più verosimili ed analizzabili, sono stati presi in considerazione aspetti quali:

- **Differenza nella probabilità di acquisto di prodotti diversi** (Ad esempio, i prodotti di classe cancelleria sono richiesti più frequentemente rispetto a quelli di classe mobilia)
- **Differenze nei costi dei prodotti sulla base della classe merceologica** (Ad esempio, i prodotti della classe elettronica hanno costi mediamente più alti rispetto a quelli della classe cancelleria)
- **Specializzazione dei fornitori** (Si prevede che alcuni fornitori siano specializzati nella vendita di articoli appartenenti ad un sottoinsieme delle classi merceologiche precedentemente definite. Tuttavia, si considerano anche fornitori il cui listino contiene articoli appartenenti a tutte le classi merceologiche)

Al fine di definire inserimenti validi, nel corso della generazione dei dati vengono, inoltre, presi in considerazione i vincoli imposti sulla base di dati e controllati dai trigger definiti in precedenza. Il periodo di attività dell'ente preso in considerazione è quello di un ipotetico anno solare (nella fattispecie, l'anno 2020).

I file vengono, infine, generati all'interno della directory `psqlOnDocker/sql`.

6.4 Generazione della base di dati

Al fine di agevolare il processo di creazione e popolamento della base di dati, è stato definito un Makefile che permette, una volta istanziato il container (con il comando `docker compose up -d`):

- La generazione dei dati di mockup (`make mockup`)¹
- La creazione e il popolamento della base di dati (`make db`)

¹La cartella `psqlOnDocker/sql` presenta al suo interno i dati di mockup già prodotti e utilizzati per le operazioni di testing.

6.5 Query significative

Sulla base delle operazioni frequenti individuate al punto 2.4 e al fine di agevolare le interrogazioni alla base di dati, vengono di seguito descritte le query significative in linguaggio SQL.

6.5.0.1 Visualizzazione di tutti gli articoli

```
-- Tutti gli articoli
SELECT * FROM Articolo;

-- Articoli filtrati per classe
SELECT * FROM Articolo WHERE Classe='cancelleria';

-- Articoli filtrati per descrizione
SELECT * FROM ARTICOLO WHERE Descrizione LIKE '%penna%';

-- Articoli filtrati per descrizione, classe e unità di misura
SELECT *
FROM Articolo
WHERE Descrizione LIKE '%stampante%'
      AND Classe='informatica'
      AND UnitaDiMisura='cad';
```

6.5.0.2 Visualizzazione di tutti gli articoli con specifiche relative ai fornitori

```
SELECT Articolo.Codice,
       Articolo.Descrizione,
       Articolo.UnitaDiMisura,
       Fornisce.PrezzoUnitario,
       Fornisce.Sconto,
       Fornisce.QuantitaMinima,
       Fornitore.PartitaIVA
FROM Articolo INNER JOIN Fornisce ON Articolo.Codice=Fornisce.Articolo
      INNER JOIN Fornitore ON Fornitore.PartitaIVA=Fornisce.Fornitore
ORDER BY Articolo.Codice ASC;
```

6.5.0.3 Visualizzazione di tutti gli articoli non forniti da alcun fornitore

```
SELECT Articolo.Codice,
       Articolo.Descrizione,
       Articolo.UnitaDiMisura
FROM Articolo LEFT JOIN Fornisce ON Articolo.Codice=Fornisce.Articolo
WHERE Fornitore IS NULL;
```

6.5.0.4 Aggiornamento dello stato di un ordine

```
UPDATE Ordine SET Stato='spedito' WHERE Codice=2;

UPDATE Ordine SET Stato='consegnato' WHERE Codice=10;
```

6.5.0.5 Visualizzazione delle informazioni relative ad una Richiesta d'Acquisto

```
-- Selezione per dipartimento e numero della richiesta
SELECT *
FROM RichiestaAcquisto
WHERE Dipartimento='SIJTBK'
      AND Numero=1;

-- Selezione in un intervallo di tempo
SELECT *
FROM RichiestaAcquisto
WHERE DataEmissione BETWEEN '2020-10-01' AND '2020-11-02'
ORDER BY DataEmissione DESC;
```

6.5.0.6 Visualizzazione di tutti gli articoli contenuti in una Richiesta d'Acquisto

```
-- Selezione per dipartimento e numero della richiesta
SELECT *
FROM Include
WHERE Dipartimento='SIJTBK'
      AND NumeroRichiesta=1;

-- Selezione in un intervallo di tempo
SELECT *
FROM RichiestaAcquisto INNER JOIN Include
      ON RichiestaAcquisto.Dipartimento = Include.Dipartimento
      AND RichiestaAcquisto.Numero = Include.NumeroRichiesta
WHERE DataEmissione BETWEEN '2020-10-01' AND '2020-11-02';

-- Con informazioni relative al rispettivo ordine per ogni articolo
SELECT * FROM Include
LEFT JOIN Ordine
ON Include.Ordine=Ordine.Codice;
```

6.5.0.7 Inserimento di un nuovo ordine L'operazione di inserimento di un nuovo Ordine richiede la creazione dello stesso e, successivamente, l'aggiornamento dell'attributo **Ordine** nell'entità *Include* per tutte le entry interessate. È stata, pertanto, definita la funzione *NuovoOrdine* che, acquisendo parametri relativi al **fornitore dell'ordine** e all'insieme delle triple (*Articolo*, *NumeroRichiesta*, *Dipartimento*), costruisce un nuovo Ordine associando gli articoli specificati.

```
create or replace function InserisciOrdine(fornitore char(16), articolo integer[],
                                           richiesta integer[], dipartimento text[])
    returns void
    language plpgsql as
$$
declare
    codice integer;
begin
    if array_length(articolo, 1) = 0 then
        raise exception 'Ogni vettore deve contenere almeno un elemento';
    end if;

    if array_length(articolo, 1) = array_length(richiesta, 1) AND
       array_length(richiesta, 1) = array_length(dipartimento, 1) then

        INSERT INTO Ordine(Fornitore) VALUES (NuovoOrdine.Fornitore);

        codice = currval('ordine_codice_seq');

        UPDATE Include i
        SET Ordine = codice
        FROM (
            SELECT UNNEST(Dipartimento) as Dipartimento,
                  UNNEST(Richiesta) as NumeroRichiesta,
                  UNNEST(Articolo) as Articolo
        ) u
        WHERE i.Dipartimento = u.Dipartimento
              AND i.NumeroRichiesta = u.NumeroRichiesta
              AND i.Articolo = u.Articolo;

    else

        raise exception 'Gli array hanno cardinalità diverse';

    end if;
end;
$$;
```

6.5.0.8 Inserimento di una nuova Richiesta d'Acquisto Analogamente alla procedura di inserimento di un nuovo ordine, si è scelto di definire una funzione per l'inserimento di una Richiesta d'Acquisto. Questa permette l'inserimento della richiesta nell'entità *RichiestaAcquisto* e dei rispettivi articoli richiesti nell'entità *Include*.

```
create or replace function InserisciRichiesta(dip char(6), articolo integer[], quantita integer[])
    returns void
    language plpgsql as
$$
declare
    codice integer;
begin
    SELECT ProssimoNumero INTO Codice FROM ProssimoCodiceRichiesta WHERE Dipartimento=dip;

    if array_length(articolo, 1) IS NULL then

        raise exception 'Specificare almeno un articolo';

    elseif array_length(quantita, 1) IS NULL then

        INSERT INTO RichiestaAcquisto(Dipartimento) VALUES (dip);

        INSERT INTO Include(Dipartimento, NumeroRichiesta, Articolo, Quantita)
        SELECT dip, codice, unnest(articolo), 1;

    elseif array_length(articolo, 1) = array_length(quantita, 1) then

        INSERT INTO RichiestaAcquisto(Dipartimento) VALUES (dip);

        INSERT INTO Include(Dipartimento, NumeroRichiesta, Articolo, Quantita)
        SELECT dip, codice, unnest(articolo), unnest(quantita);
    else

        raise exception 'Gli array hanno cardinalità diverse';

    end if;
end;
$;
```

6.5.0.9 Calcolo della spesa mensile dei dipartimenti Si definisce la query che, dato un intervallo di tempo espresso tramite **data di inizio** e **data di fine**, calcola, per ogni dipartimento, il numero di richieste d'acquisto effettuate e la spesa complessiva.

```
SELECT i.Dipartimento AS "Dipartimento",
       COUNT(DISTINCT NumeroRichiesta) AS "Richieste",
       SUM(PrezzoUnitario*Quantita) AS "Spesa"
FROM Include AS i INNER JOIN RichiestaAcquisto AS r
     ON r.dipartimento = i.dipartimento AND r.numero = i.numerorichiesta
WHERE DataEmissione BETWEEN '2021-01-01' AND '2021-02-01'
GROUP BY i.Dipartimento;
```

6.5.0.10 Calcolo della spesa complessiva dell'ente in un intervallo di tempo

```
SELECT COUNT(DISTINCT NumeroRichiesta) AS "Richieste",
       SUM(PrezzoUnitario*Quantita) AS "Spesa"
FROM Include AS i INNER JOIN RichiestaAcquisto AS r
     ON r.dipartimento = i.dipartimento AND r.numero = i.numerorichiesta
WHERE DataEmissione BETWEEN '2021-01-01' AND '2021-02-01';
```

7 Analisi dei dati

In seguito all'implementazione della base di dati e all'inserimento dei dati di mockup appositamente generati, è stato possibile produrre un'analisi dei dati con rispettive visualizzazioni grafiche a partire da opportune interrogazioni in linguaggio SQL.

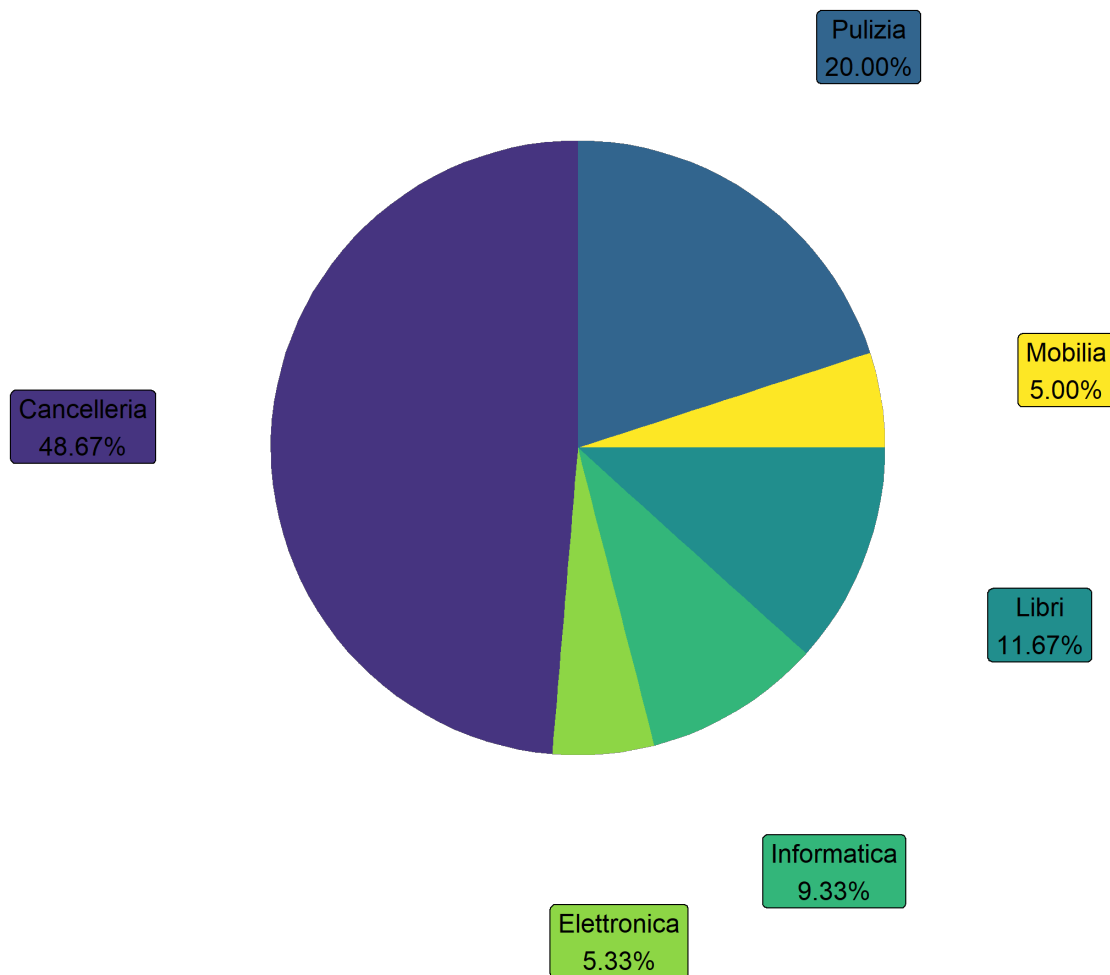
A tal fine, è stato prodotto un notebook in linguaggio **R Markdown** situato al percorso file (`R/DataAnalysis.Rmd`) che utilizza la libreria [RPostgreSQL](#) assieme ad ulteriori librerie quali [dplyr](#) e [ggplot2](#) per la manipolazione dei dati e la produzione di opportune visualizzazioni.

Per una migliore visualizzazione, i grafici ad alta risoluzione sono disponibili al percorso file `R/analysisPlots`.

7.1 Distribuzione delle classi merceologiche

A partire dalla seguente interrogazione è stato possibile visualizzare la distribuzione di tutti gli articoli sulla base della loro classe merceologica. Come atteso, sulla base delle modalità di produzione dei dati di mockup impiegate, si osserva una prevalenza degli articoli di **cancelleria**.

```
SELECT Classe,  
       COUNT(*)/(SUM(COUNT(*)) OVER()) AS Frequenza  
FROM Articolo  
GROUP BY Classe;
```

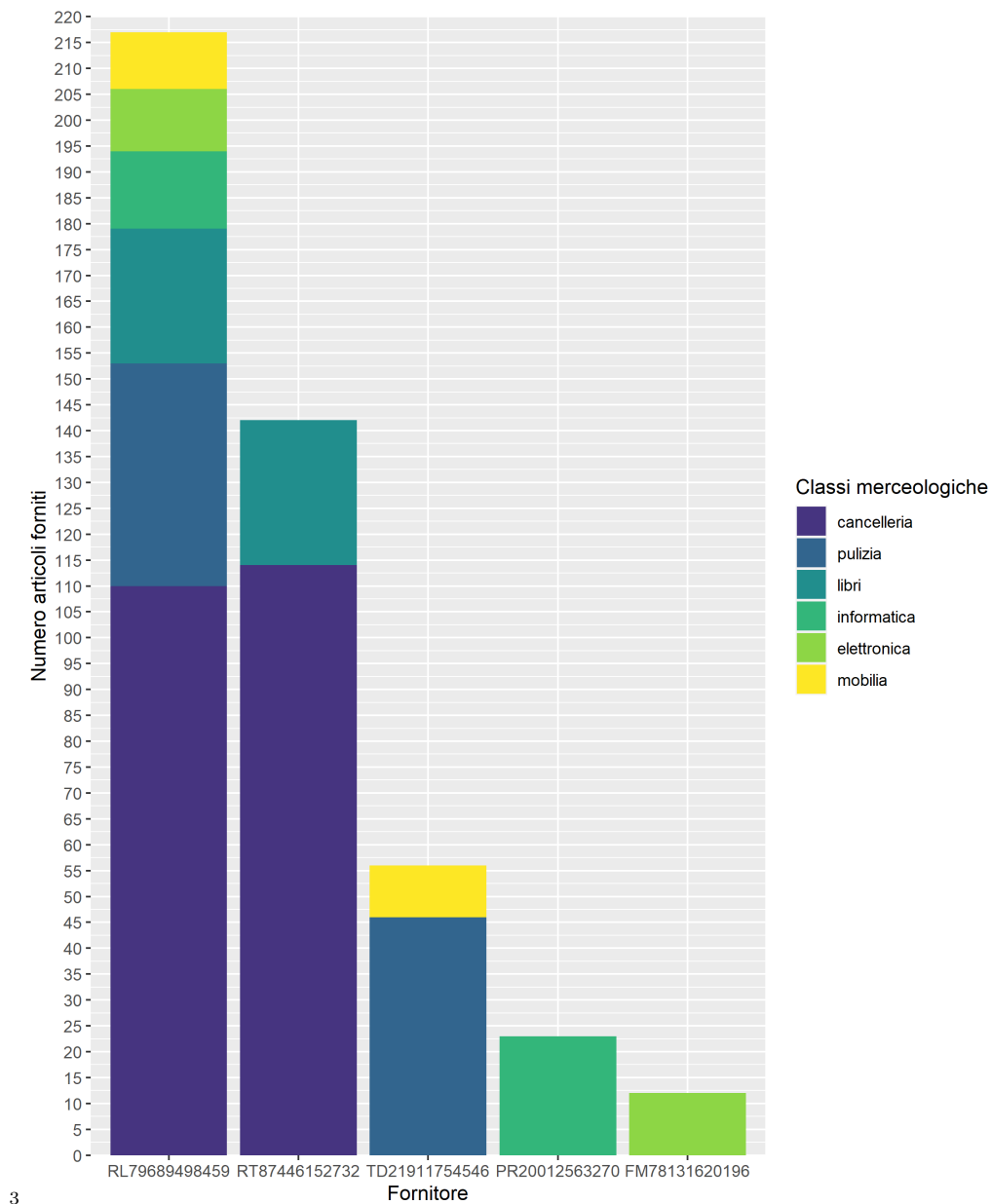


2

7.2 Distribuzione degli articoli per ogni fornitore

A partire dall'interrogazione seguente, è stato prodotto un barplot atto a raffigurare la distribuzione degli articoli forniti da ognuno dei fornitori, con un'ulteriore suddivisione basata sulle diverse classi merceologiche.

```
SELECT Fornitore, Classe, COUNT(*) AS Frequenza
FROM Fornisce f
  JOIN (SELECT Codice, Classe FROM Articolo) a ON f.Articolo = a.Codice
GROUP BY Fornitore, Classe;
```

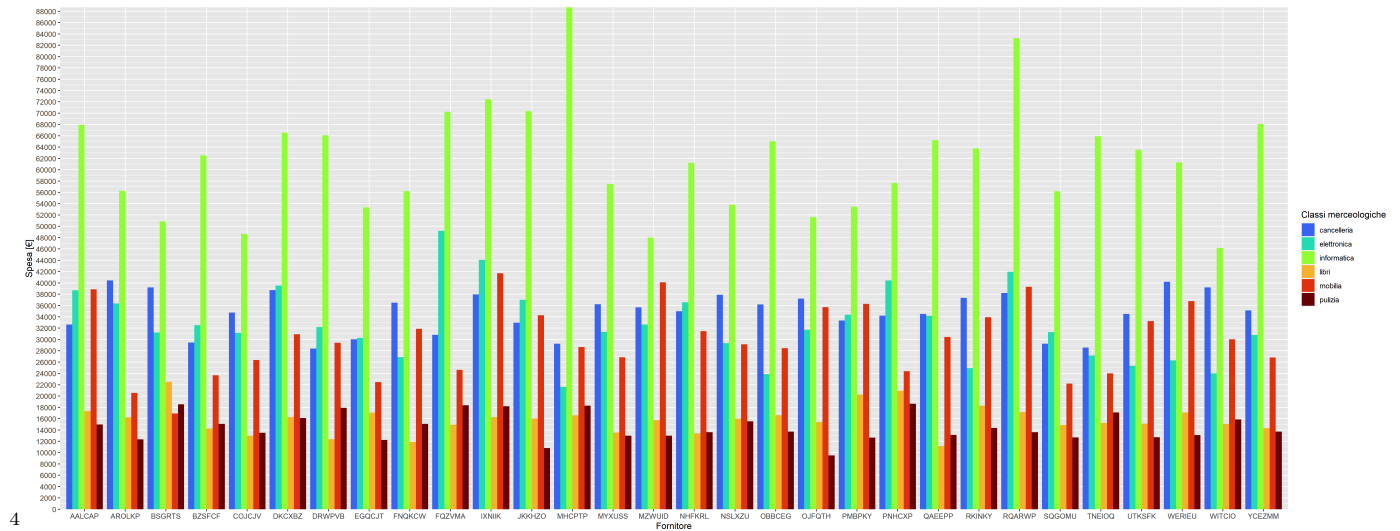


³R/analysisPlots/distribuzione_articoli_fornitore.png

7.3 Confronto della spesa dei dipartimenti

A partire dall'interrogazione seguente, è stato prodotto un barplot atto a raffigurare, per ogni dipartimento, la spesa effettuata per articoli appartenenti alle varie classi merceologiche definite, nell'anno solare considerato.

```
SELECT Dipartimento, Classe, SUM(Quantita * Prezzounitario) AS SPESA
FROM Include i
JOIN (SELECT Codice, Classe FROM Articolo) a ON i.Articolo = a.Codice
GROUP BY Dipartimento, Classe;
```

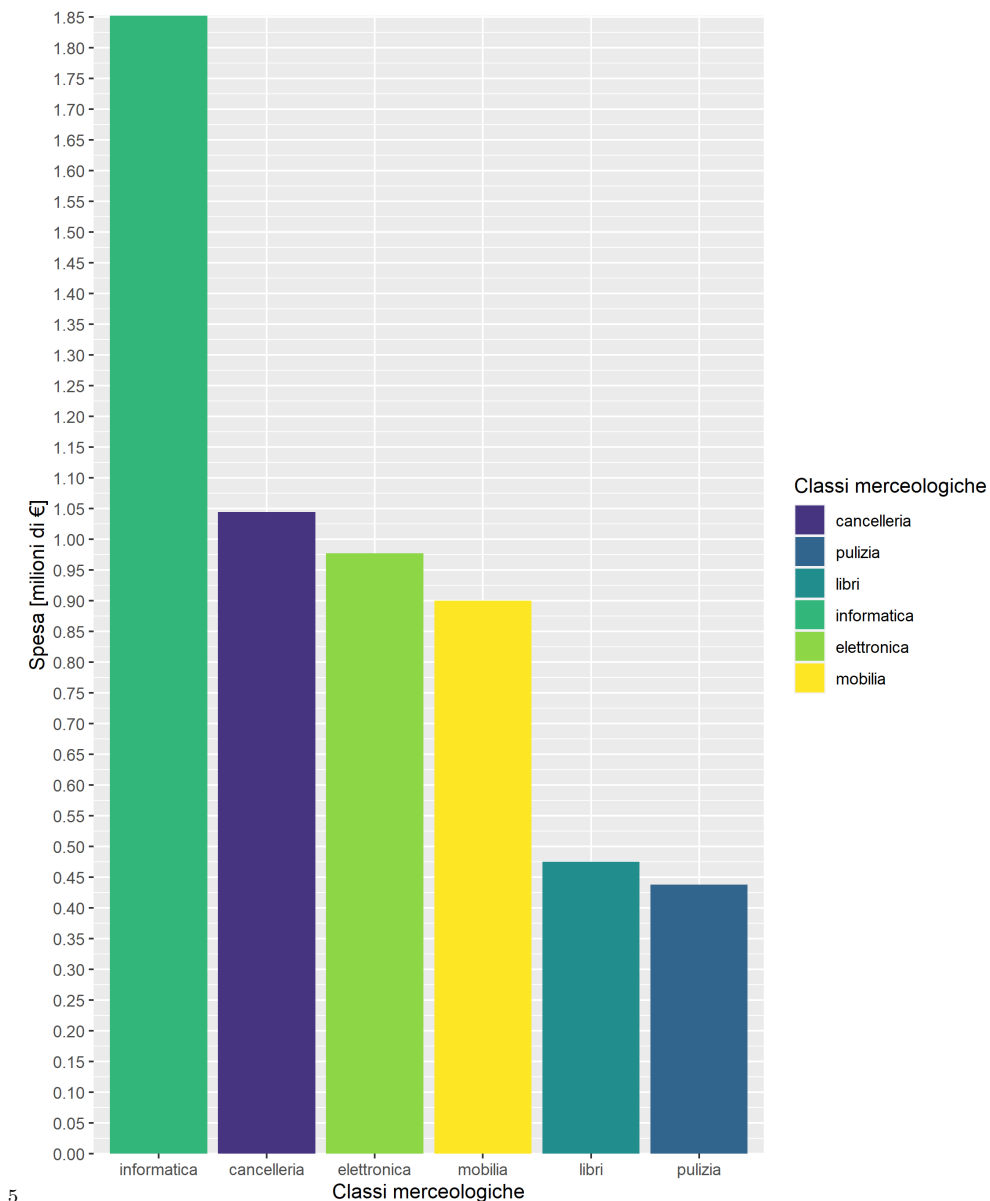


⁴R/analysisPlots/spesa_dipartimento_classe.png

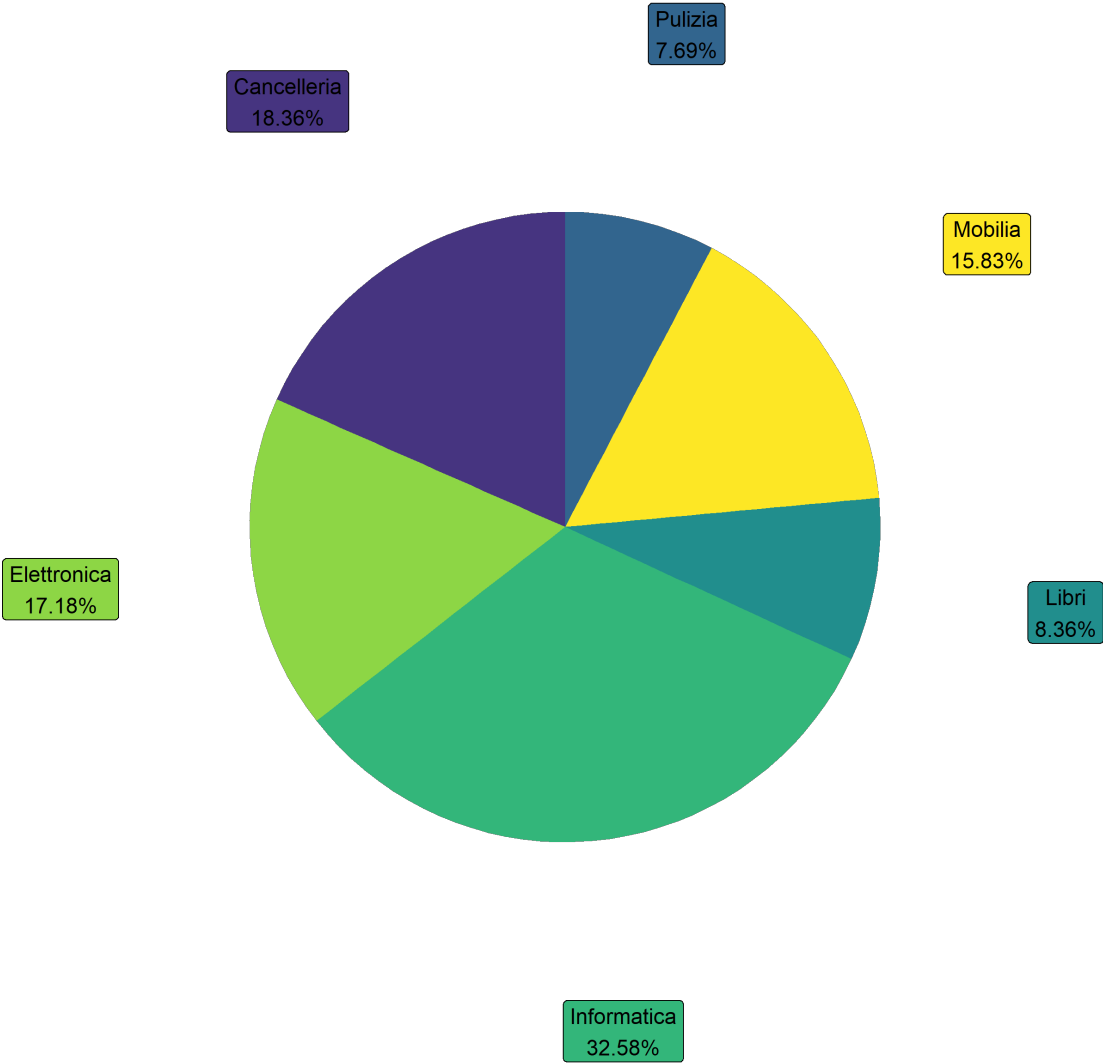
7.4 Spesa totale per classe merceologica

A partire dall'interrogazione seguente, è stato prodotto un barplot che raffigura la spesa complessiva, nel corso dell'anno solare considerato, per ognuna delle classi merceologiche. Si osserva come i prodotti di classe *informatica* siano quelli che hanno richiesto la spesa maggiore, mentre articoli di altre classi hanno comportato una spesa più simile fra loro. Ciò è ragionevole immaginando che prodotti appartenenti alla classe informatica abbiano un costo maggiore rispetto ad articoli di, ad esempio, cancelleria.

```
SELECT Classe,  
       SUM((Quantita * PrezzoUnitario)/1000000) AS Spesa,  
       SUM(  
         (Quantita * PrezzoUnitario)/1000000) /  
         (SUM(SUM((Quantita * PrezzoUnitario)/1000000)  
          ) OVER()) AS Frequenza  
FROM Include i  
   JOIN (SELECT Codice, Classe FROM Articolo) a ON i.articolo = a.codice  
GROUP BY Classe;
```



È stato, inoltre, prodotto un diagramma a torta basato sulla stessa interrogazione SQL, che rappresenta le frequenze relative delle spese effettuate per ognuna delle classi merceologiche.



6

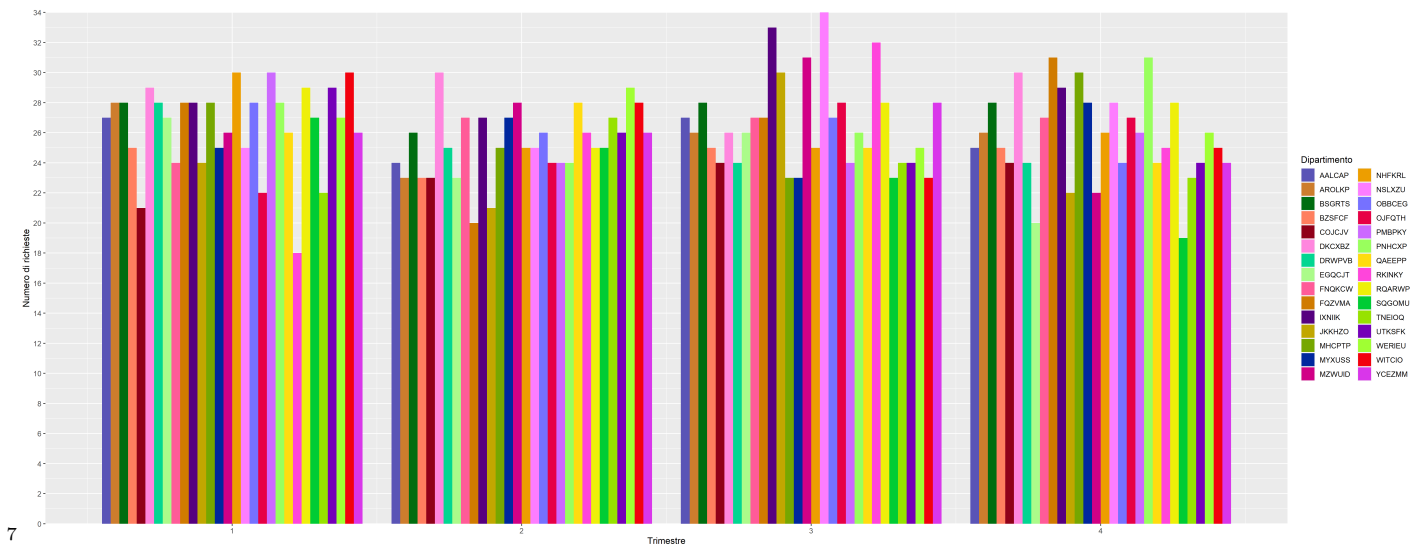
⁵R/analysisPlots/spesa_classe.png

⁶R/analysisPlots/spesa_classe_pie.png

7.5 Richieste d'acquisto trimestrali effettuate dai dipartimenti

A partire dall'interrogazione seguente, è stato prodotto un barplot atto a raffigurare, per ognuno dei dipartimenti, il numero di richieste d'acquisto effettuate trimestralmente.

```
SELECT Dipartimento, COUNT(*) NumeroRichieste, CASE
  WHEN EXTRACT(MONTH FROM DataEmissione) < 4 THEN 1
  WHEN EXTRACT(MONTH FROM dataemissione) < 7 THEN 2
  WHEN EXTRACT(MONTH FROM dataemissione) < 10 THEN 3
  ELSE 4 END Trimestre
FROM RichiestaAcquisto
GROUP BY Dipartimento, Trimestre
ORDER BY Trimestre, NumeroRichieste;
```

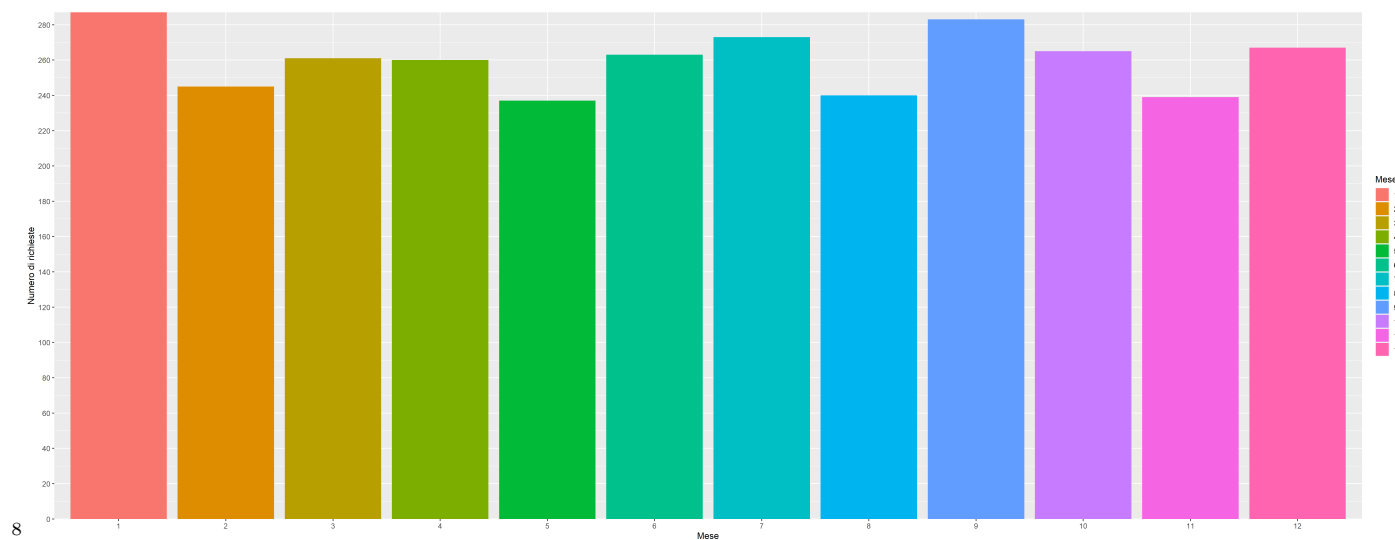


⁷R/analysisPlots/ricieste_dipartimento_trimestre.png

7.6 Numero di richieste d'acquisto mensili

A partire dall'interrogazione seguente, è stato prodotto un barplot atto a raffigurare la quantità di richieste d'acquisto effettuate per ogni mese dell'anno solare. Si osserva, in particolare, come i mesi di gennaio e settembre siano stati quelli con maggior numero di richieste.

```
SELECT EXTRACT(MONTH FROM DataEmissione) Mese, COUNT(*) NumeroRichieste
FROM RichiestaAcquisto
GROUP BY Mese
ORDER BY Mese;
```

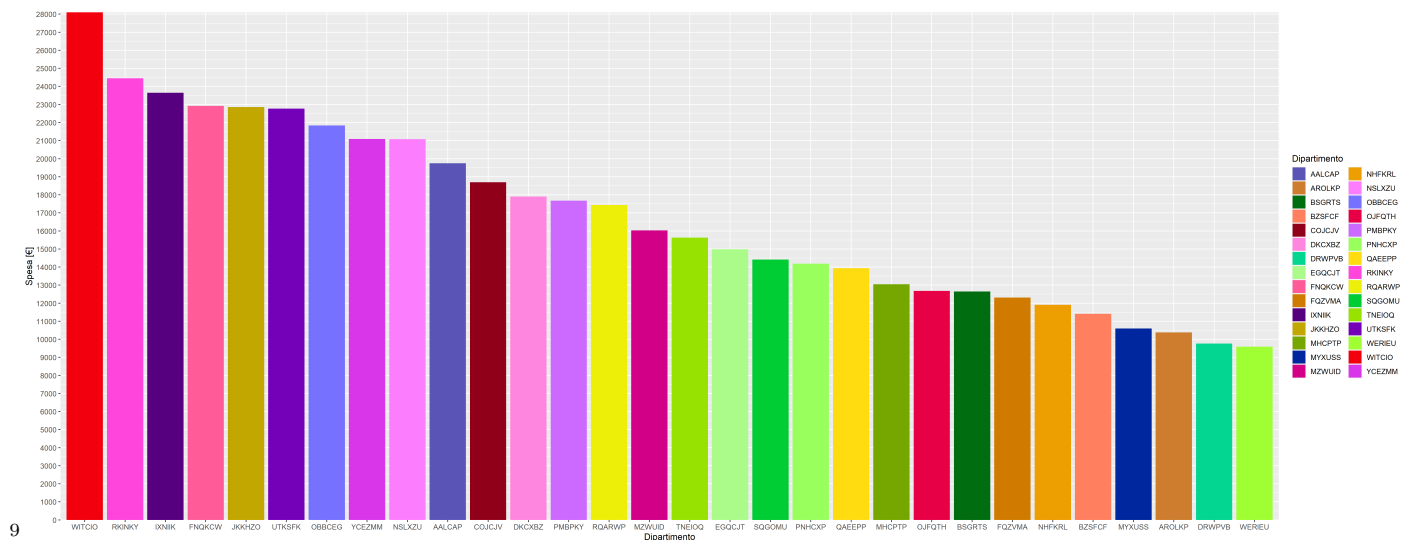


⁸R/analysisPlots/richieste_mensili.png

7.7 Spesa dei dipartimenti nel mese di giugno

A partire dall'interrogazione seguente, è stato prodotto un barplot che raffigura la spesa effettuata da ogni dipartimento nel corso del mese di giugno. Si osserva, in particolare, come il dipartimento WITC10 sia quello che ha richiesto la spesa maggiore. Modificando opportunamente la condizione della query SQL, è possibile riprodurre il diagramma per qualunque altro mese dell'anno solare.

```
SELECT i.Dipartimento, SUM((Quantita * PrezzoUnitario)) Spesa
FROM Include i
      JOIN (SELECT Dipartimento, Numero, DataEmissione FROM RichiestaAcquisto) r
      ON r.Dipartimento = i.Dipartimento AND r.Numero = i.NumeroRichiesta
WHERE EXTRACT(MONTH FROM DataEmissione) = 6
GROUP BY i.Dipartimento
ORDER BY i.Dipartimento;
```

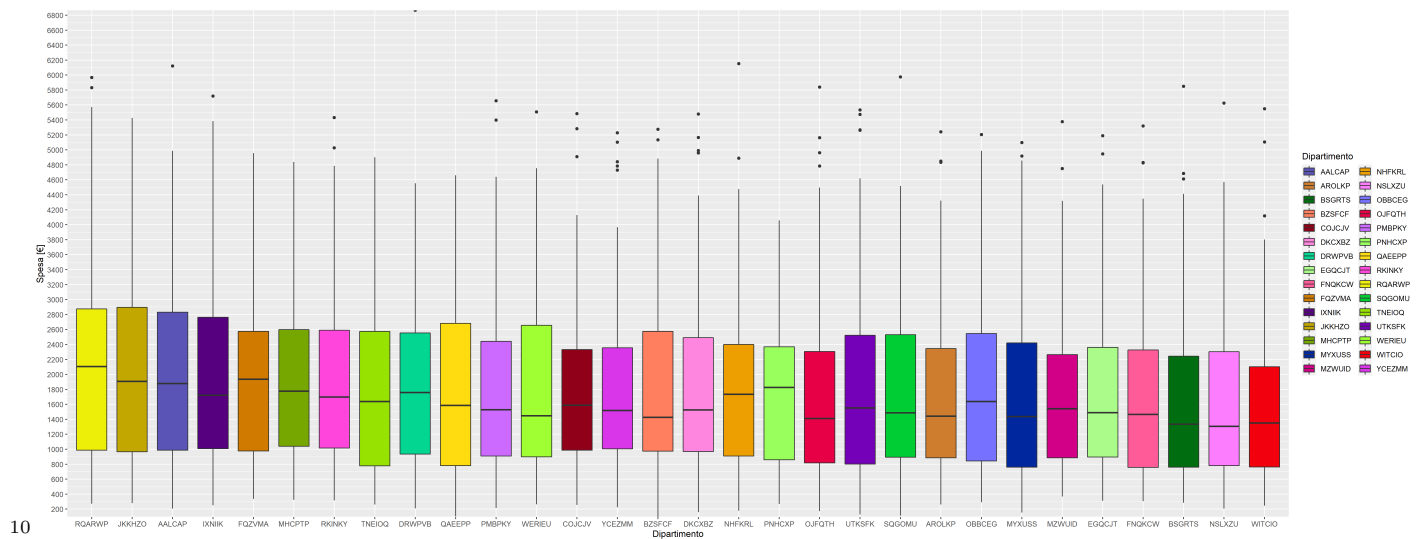


⁹R/analysisPlots/spesa_dipartimento_giugno.png

7.8 Spesa giornaliera dei dipartimenti

A partire dall'interrogazione seguente, è stato prodotto un boxplot che raffigura la distribuzione della spesa giornaliera da parte di ogni dipartimento.

```
SELECT i.Dipartimento, NumeroRichiesta, SUM((Quantita * PrezzoUnitario)) Spesa
FROM Include i
  JOIN (SELECT Dipartimento, Numero, DataEmissione FROM RichiestaAcquisto) r
  ON r.Dipartimento = i.Dipartimento AND r.Numero = i.NumeroRichiesta
GROUP BY i.Dipartimento, i.NumeroRichiesta
ORDER BY i.Dipartimento, i.NumeroRichiesta;
```



¹⁰R/analysisPlots/spesa_giornaliera_dipartimenti.png

8 Conclusioni

Il presente elaborato ha permesso la descrizione dell'attività di progettazione e implementazione di una base di dati relazionale a partire da un insieme di requisiti e specifiche. Sulla base dei pattern progettuali studiati, sono state affrontate le fasi di Analisi dei Requisiti, Progettazione Concettuale, Progettazione Logica e la successiva Progettazione Fisica con implementazione tramite **PostgreSQL**. Infine, tramite il linguaggio **R**, è stato interrogato il DBMS al fine di produrre opportune visualizzazioni e statistiche riassuntive atte ad analizzare i dati in esso contenuti.