



**UNIVERSITÀ
DEGLI STUDI
DI UDINE**

Progetto per la realizzazione di una base di dati aziendale

Antonutti Matteo - matr. 153259

Da Frè Tommaso - matr. 152744

Fioritto Alessandro - matr. 152205

Londero Alessandro - matr. 152102

Zampieri Giacomo - matr. 153321

LINK REPOSITORY GITHUB

<https://github.com/GiacomoZampieri/Progetto-Basi-di-Dati>

1 RICHIESTA	2
1.1 Fasi del progetto	3
2 ANALISI DEI REQUISITI	3
2.1 Glossario dei termini	3
2.2 Regole relazionali e vincoli d'integrità	4
2.3 Operazioni sulla base di dati	4
2.4 Caratteristiche delle singole entità	5
3 PROGETTAZIONE CONCETTUALE	6
3.1 Diagrammi E-R delle singole entità	6
3.2 Diagramma E-R	8
4 PROGETTAZIONE LOGICA	9
4.1 Analisi delle ridondanze	9
4.1.1 Tabella volumi	10
4.1.2 Tabella frequenze	11
4.1.3 Tabella accessi	11
4.1.4 Conclusioni sulla possibile aggiunta della ridondanza	13
4.1.5 Parametri richiesti ma non presenti come attributi	14
4.2 Eliminazione delle generalizzazioni	15
4.3 Scelta delle chiavi primarie	15
4.4 Schema E-R dopo la progettazione logica	16
4.5 Traduzione da schema E-R a logico	17
4.6 Schema logico	17
5 PROGETTAZIONE FISICA	18
5.1 Domini	18
5.2 Creazione tabelle	18
6 IMPLEMENTAZIONE	21
6.1 Popolamento del DataBase	21
7 ANALISI DEI DATI IN R	24

1 RICHIESTA

Si vuole progettare una base di dati per la gestione del personale di un'azienda.

L'azienda è organizzata in un insieme di dipartimenti ed ogni dipartimento è caratterizzato da un insieme di impiegati, un insieme di progetti e un insieme di uffici.

Ogni impiegato ha una storia lavorativa (insieme dei lavori svolti in passato); inoltre, per ciascuno di tali lavori, l'impiegato ha una storia retributiva (insieme degli stipendi percepiti nel periodo durante il quale ha svolto un certo lavoro).

Ogni ufficio ha un insieme di telefoni.

La base di dati dovrà contenere le seguenti informazioni:

- Per ogni dipartimento, il numero, il nome, il budget e il codice dell'impiegato responsabile.
- Per ogni impiegato, il codice dell'impiegato, il numero del progetto cui sta attualmente lavorando, il numero dell'ufficio e il numero del telefono a lui assegnato (si assuma che ogni impiegato disponga di un telefono e che ogni telefono sia assegnato ad un impiegato). Inoltre, per ogni lavoro svolto in passato, il tipo di lavoro svolto, più la data di inizio, la data di fine e l'ammontare complessivo del compenso ricevuto per tale lavoro.
- Per ogni progetto, il numero del progetto, il numero del dipartimento responsabile del progetto e il budget; il budget di ciascun dipartimento deve essere maggiore o uguale alla somma dei budget dei progetti da esso coordinati.
- Per ogni ufficio, il numero dell'ufficio (che identifica univocamente l'ufficio nell'ambito del dipartimento cui appartiene; ad esempio, ufficio 23 del dipartimento 5), il piano dove si trova (si assuma che tutti gli uffici di tutti i dipartimenti si trovino in uno stesso edificio a più piani) e i numeri di tutti i telefoni presenti nell'ufficio. Si assuma, infine, che più impiegati possano condividere uno stesso ufficio.

Si definisca uno schema Entità-Relazioni che descriva il contenuto informativo del sistema, illustrando con chiarezza le eventuali assunzioni fatte. Lo schema dovrà essere completato con attributi ragionevoli per ciascuna entità (identificando le possibili chiavi) e relazione. Vanno specificati accuratamente i vincoli di cardinalità e partecipazione di ciascuna relazione.

1.1 Fasi del progetto

Di seguito sono illustrate le fasi seguite nella realizzazione del progetto

- Analisi della richiesta e dei requisiti;
- Prima fase di progettazione (concettuale);
- Seconda fase di progettazione (logica) e conseguenti correzioni sulla prima fase;
- Terza fase di progettazione (fisica);
- Implementazione effettiva;
- Analisi dei dati tramite R e conclusioni;

2 ANALISI DEI REQUISITI

2.1 Glossario dei termini

Termine	Descrizione	Collegamenti
Dipartimento	Il dipartimento è un insieme di dipendenti che si adopererà su specifici progetti, i quali sono locati in corrispondenti uffici. L'azienda è divisa in dipartimenti.	Ufficio Progetto Impiegato Responsabile
Dipendente	Lavora nell'azienda; si divide in impiegato e responsabile; il responsabile dirige un dipartimento, mentre un impiegato ci lavora.	Dipartimento Lavoro
Ufficio	L'ufficio è un luogo di lavoro locato ad un certo piano di un'azienda. Uno o più uffici compongono un dipartimento.	Dipartimento
Progetto	Il progetto è un'insieme di attività portate avanti da un dipartimento per produrre un prodotto o servizio finale.	Dipartimento
Lavoro	Rappresenta tutto l'insieme dei lavori svolti da un dipendente (sia conclusi che in corso). Descrive il ruolo ricoperto in un determinato periodo con il relativo compenso ottenuto.	Dipendente

2.2 Regole relazionali e vincoli d'integrità

Le regole relazionali da rispettare durante la progettazione della base di dati sono:

- Un dipendente non può avere una storia lavorativa vuota;
- Un dipendente possiede sempre un telefono (e quel telefono è assegnato unicamente all'impiegato);
- Il budget di ciascun dipartimento deve essere maggiore o uguale alla somma dei budget dei progetti da esso coordinati;
- Un dipendente che appartiene ad un dipartimento non potrà avere il codice progetto di uno dei progetti di cui non si occupa il suo dipartimento;
- La data d'inizio di un progetto non può essere futura alla data di fine;

2.3 Operazioni sulla base di dati

Le operazioni principali da operare sulla base di dati sono:

- Ottenere la storia lavorativa di un impiegato;
- Ottenere la storia retributiva di un impiegato;
- Ottenere i numeri di telefono all'interno di un ufficio;
- Tenere traccia delle informazioni di un dipendente e dell'insieme dei lavori svolti da esso;
- Modifica del lavoro del dipendente (es. promozione o cambio dipartimento);
- Lettura del budget rimanente di un dipartimento;
- Aggiunta di un nuovo progetto;
- Modificare il budget destinato ad un singolo progetto;

2.4 Caratteristiche delle singole entità

L'entità "**Dipartimento**":

- È identificato da un nome e da un numero (univoco)
- Ha un determinato budget
- Fa riferimento ad un impiegato responsabile
- È composto da uffici
- Controlla uno o più progetti

L'entità "**Dipendente**":

- È identificato da un nome, un cognome e, univocamente, da un codice
- Ha un codice relativo al progetto al quale sta lavorando
- È assegnato ad un certo ufficio
- Ha un numero di telefono
- Ha una storia lavorativa (ricavato attraverso un'opportuna interrogazione alla tabella Lavoro)
- Ha una storia retributiva per ogni lavoro eseguito (ricavato attraverso un'opportuna interrogazione sulla tabella Lavoro)

L'entità "**Lavoro**":

- È identificato univocamente da un codice id
- È identificato da un nome, descrittivo del tipo di lavoro
- Ha una data d'inizio ed una data di fine
- Ha lo stipendio, che rappresenta il compenso mensile
- Ha un codice del dipendente che svolge quel lavoro

L'entità "**Progetto**":

- È identificato univocamente da un numero
- Ha un dipartimento responsabile che se ne occupa
- Ha un determinato budget
- Ha una data di fine stimata

L'entità "**Ufficio**":

- È identificato univocamente da un numero di ufficio e dal numero del dipartimento di cui fa parte
- Si trova ad un certo piano
- Ha tutti i numeri di telefono di tutti gli impiegati presenti nell'ufficio (ricavati attraverso un'opportuna interrogazione sulla tabella Dipendente).

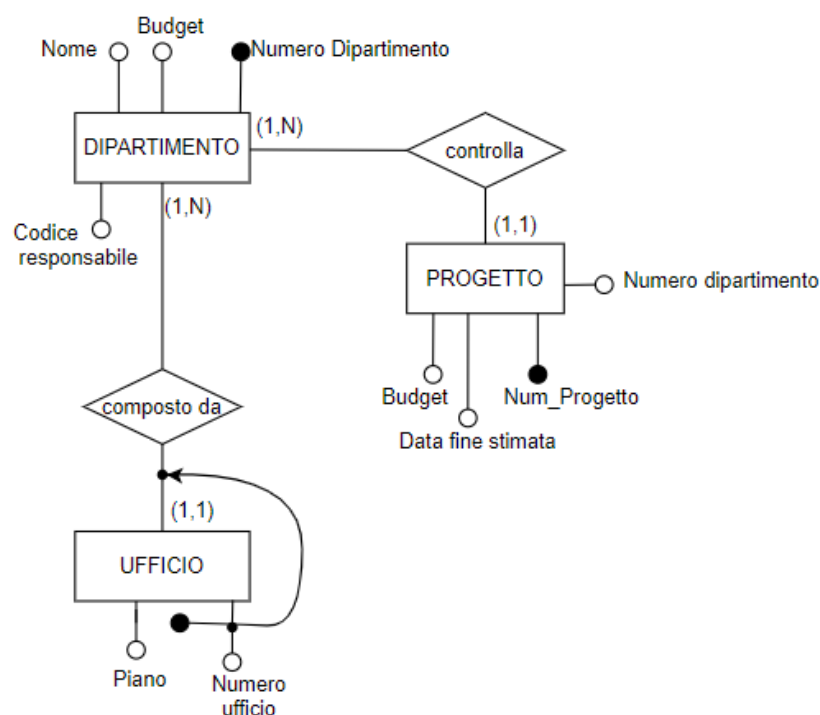
3 PROGETTAZIONE CONCETTUALE

La progettazione concettuale prevede di utilizzare i dati e le informazioni ottenute durante la raccolta/analisi dei requisiti per individuare entità e relazioni significative presenti nella base di dati per poi unirli assieme e creare una prima versione del diagramma E-R.

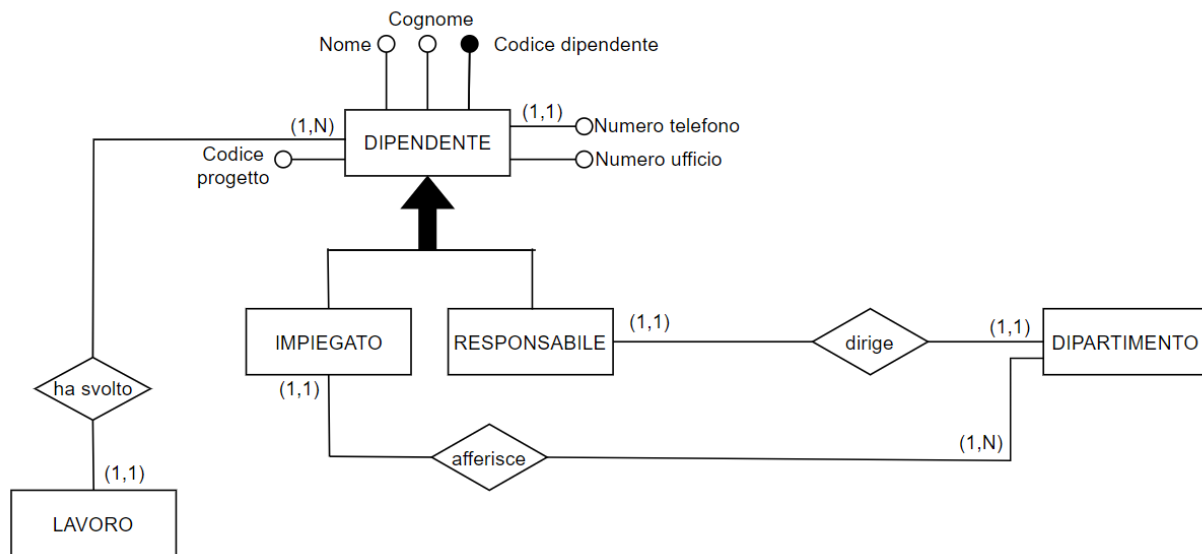
3.1 Diagrammi E-R delle singole entità

L'azienda è organizzata in più dipartimenti, composti da uno o più uffici e questi uffici fanno parte di un solo dipartimento.

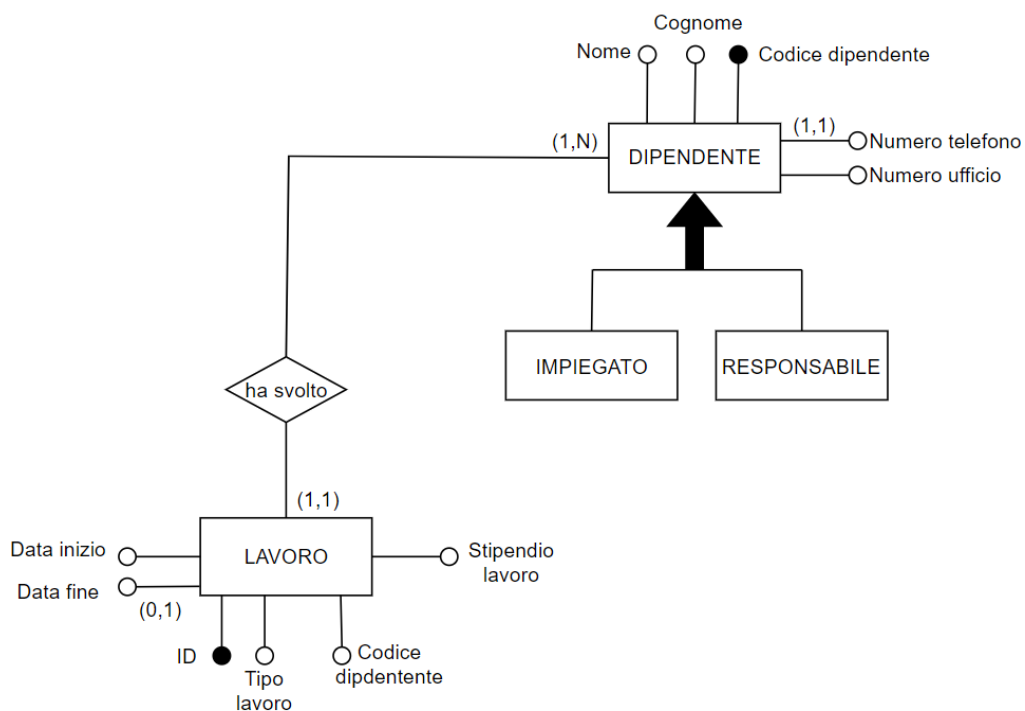
Ogni dipartimento controlla uno o più progetti.



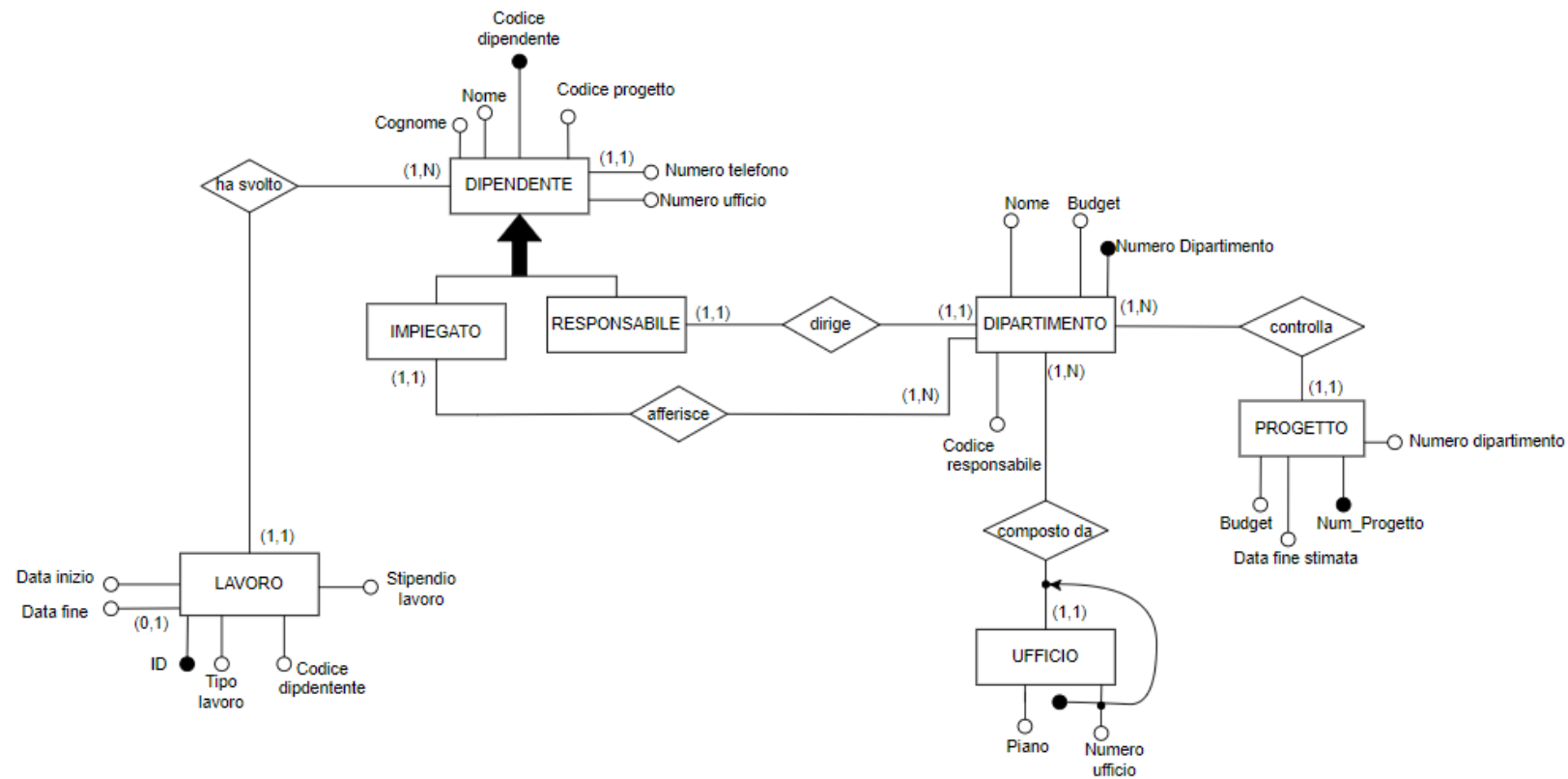
Ogni impiegato fa parte di un dipartimento.



Inoltre ha una storia lavorativa del quale si conosce la tipologia, la retribuzione totale e il periodo dell'impiego.



3.2 Diagramma E-R



4 PROGETTAZIONE LOGICA

La prima fase della progettazione logica consiste nell'analisi del diagramma E-R, non ancora ottimizzato, tenendo conto di come la progettazione concettuale impatti l'effettivo funzionamento del DataBase. Successivamente si effettueranno delle modifiche usando le seguenti tecniche:

- Analisi delle ridondanze;
- Eliminazione delle generalizzazioni;
- Partizionamento/accoppiamento delle entità e relazioni;
- Scelta delle chiavi primarie;

4.1 Analisi delle ridondanze

L'analisi delle ridondanze prevede di individuare chiaramente le ripetizioni inserite nella fase di progettazione e decidere se mantenerle o toglierle a seconda della diminuzione o dell'aumento del carico di lavoro sulla base di dati.

Nel nostro caso, un elemento cui ridondanza potrebbe risultare utile sarebbe il budget rimanente di un dipartimento.

Per verificare questa ipotesi, vengono svolte delle analisi sulle possibili operazioni di cui vogliamo studiare l'aggiunta di ridondanza. Le operazioni sono organizzate in varie tabelle che, una volta compilate, forniscono i dati necessari a trovare la risposta al quesito.

4.1.1 Tabella volumi

La tabella dei volumi permette di visualizzare quant'è indicativamente grande il dataset, listando quanti "elementi" ha ogni entità e quanti collegamenti ha ogni relazione.

Concetto	E/R	Volume	Formula
Dipartimento	E	9	Valore prestabilito
Dipendente	E	$9 \times 80 = 720$	In media 80 dipendenti per dipartimento
Responsabile	E	$1 \times 9 = 9$	1 per dipartimento
Impiegato	E	$720 - 9 = 711$	N° dipendenti totali - n° responsabili
Lavoro	E	$720 \times 1,61 = 1160$	In media 1,61 lavori per dipendente
Progetto	E	$18 \times 9 = 162$	In media 18 progetti per dipartimento
Ufficio	E	$720 / 20 = 36$	In media 20 dipendenti per ufficio
ha svolto	R	1160	Equivalente al numero di lavori
dirige	R	9	Equivalente al numero di responsabili
afferisce	R	711	Equivalente al numero di impiegati
controlla	R	162	Equivalente al numero di progetti
composto da	R	36	Equivalente al numero di uffici

Con l'introduzione dell'attributo ridondante 'budget rimanente' sono state individuate tre operazioni tra le molteplici che vengono coinvolte:

- Lettura del budget rimanente di un dipartimento;
- Aggiunta di un nuovo progetto;
- Modificare il budget destinato ad un singolo progetto;

Scegliendo di aggiungere la ridondanza come dato di tipo *float*, i **costi di memoria** conseguenti risultano essere al più:

$$4 \text{ byte} \times 9 = \mathbf{36 \text{ byte}}$$

4.1.2 Tabella frequenze

La tabella delle frequenze permette semplicemente di visualizzare la frequenza effettiva di ogni operazione.

Operazione	Tipo	Frequenza media
Lettura del budget rimanente di un dipartimento	I	10 volte al giorno
Aggiunta/rimozione di un progetto	I	1 volta ogni 2 settimane
Ridurre il budget di dipartimento	I	1 volta al mese

4.1.3 Tabella accessi

La tabella degli accessi mette a confronto il numero di accessi al database nelle tre operazioni individuate in precedenza con e senza l'utilizzo della ridondanza; inoltre permette di calcolare il numero di operazioni in più o in meno nei due casi e conseguentemente la quantità di dati che è possibile risparmiare.

Operazione 1:

Lettura del budget rimanente di un dipartimento.

Concetto	N° accessi	Tipo accesso
Con ridondanza:		
Dipartimento	1	R
Senza ridondanza:		
Dipartimento	1	R
Progetto	162	R

Nella prima operazione con l'aggiunta della ridondanza risparmieremmo $162 \cdot 10 = 1620$ operazioni al giorno.

Operazione 2:

Aggiunta di un nuovo progetto.

Concetto	N° accessi	Tipo accesso
Con ridondanza:		
Dipartimento	1	R
Progetto	1	W
Dipartimento	1	W
Senza ridondanza:		
Progetto	162	R
Dipartimento	1	R
Progetto	1	W

Nella seconda operazione con l'aggiunta della ridondanza risparmieremmo 161 operazioni ogni 2 settimane.

Operazione 3:

Modificare il budget destinato ad un singolo progetto.

Concetto	N° accessi	Tipo accesso
Con ridondanza:		
Dipartimento	1	R
Progetto	1	W
Dipartimento	1	W
Senza ridondanza:		
Progetto	162	R
Dipartimento	1	R
Progetto	1	W

Nella terza operazione con l'aggiunta della ridondanza risparmieremmo 161 operazioni ogni mese.

4.1.4 Conclusioni sulla possibile aggiunta della ridondanza

Operazione	Letture	Scritture
Con ridondanza:		
Operazione 1	$1 \times 10 \times 20 = 200$	0
Operazione 2	1×2	2×2
Operazione 3	1	2
TOTALE:	203	6
Senza ridondanza:		
Operazione 1	$163 \times 10 \times 20 = 32600$	0
Operazione 2	$163 \times 2 = 326$	1×2
Operazione 3	163	1
TOTALE:	33089	3

Dalla tabella, considerando un arco temporale di un mese, si evince che con l'aggiunta della ridondanza si andrebbero a risparmiare circa 32886 letture per operazione ogni mese; è giusto citare che le scritture raddoppierebbero, nonostante l'aumento sia sostanzialmente trascurabile (da 3 a 6).

In conclusione, è stato deciso di aggiungere la ridondanza, visto che il risparmio di risorse è evidentemente superiore allo spazio occupato dal dato.

4.1.5 Parametri richiesti ma non presenti come attributi

Per determinati parametri richiesti dal progetto, è stato scelto di non inserirli come attributi poiché ridondanti e ottenibili da informazioni già presenti nel database.

Tali informazioni sono ottenibili tramite opportune query. Nello specifico riguardano le seguenti richieste:

Numeri di telefono di un ufficio:

Per rispondere a questa richiesta non è stato realizzato un parametro ad-hoc per l'entità ufficio, poiché è stata fatta l'assunzione secondo la quale i numeri di telefono di un ufficio corrispondono esattamente ai soli numeri di telefono dei dipendenti che vi lavorano.

Per sapere l'insieme di numeri di telefono di uno specifico ufficio, appartenente al dipartimento n e avente numero di ufficio m è sufficiente la seguente query al database:

```
SELECT D.telefono, U.dipartimento, D.ufficio, D.codice_dipendente  
FROM dipendente AS D, ufficio AS U  
WHERE D.ufficio = U.numero_ufficio AND U.dipartimento =  $n$  AND D.ufficio =  $m$ 
```

Storia lavorativa di un dipendente:

Essendo la storia lavorativa l'insieme di lavori svolti (o in corso di svolgimento), per risalire a questa informazione relativa ad un dipendente avente codice dipendente cd è sufficiente la seguente interrogazione alla base di dati:

```
SELECT L.impiegato, D.nome, D.cognome, L.data_inizio, L.data_fine, L.stipendio_lavoro  
FROM lavoro AS L, dipendente AS D  
WHERE L.impiegato = D.codice_dipendente and L.impiegato =  $cd$ 
```

Storia retributiva di un dipendente:

La storia retributiva corrisponde all'ammontare complessivo degli stipendi percepiti da un dipendente nel periodo nel quale ha svolto un certo lavoro.

Assumendo che la retribuzione annuale sia distribuita su 14 mensilità, per ottenere tale informazione (relativa al lavoro avente id id_lavoro di un dipendente):

```
SELECT D.nome, D.cognome, L.tipo_lavoro, L.id, L.data_inizio,  
L.data_fine, (((DATE_PART('year', L.data_fine::timestamp) - DATE_PART('year',  
L.data_inizio::timestamp)) * 14 + (DATE_PART('month', L.data_fine::timestamp) -  
DATE_PART('month', data_inizio::timestamp))) * L.stipendio_lavoro) AS ammontare  
FROM lavoro AS L, dipendente AS D  
WHERE L.impiegato = D.codice_dipendente AND L.impiegato =  $cd$ 
```

4.2 Eliminazione delle generalizzazioni

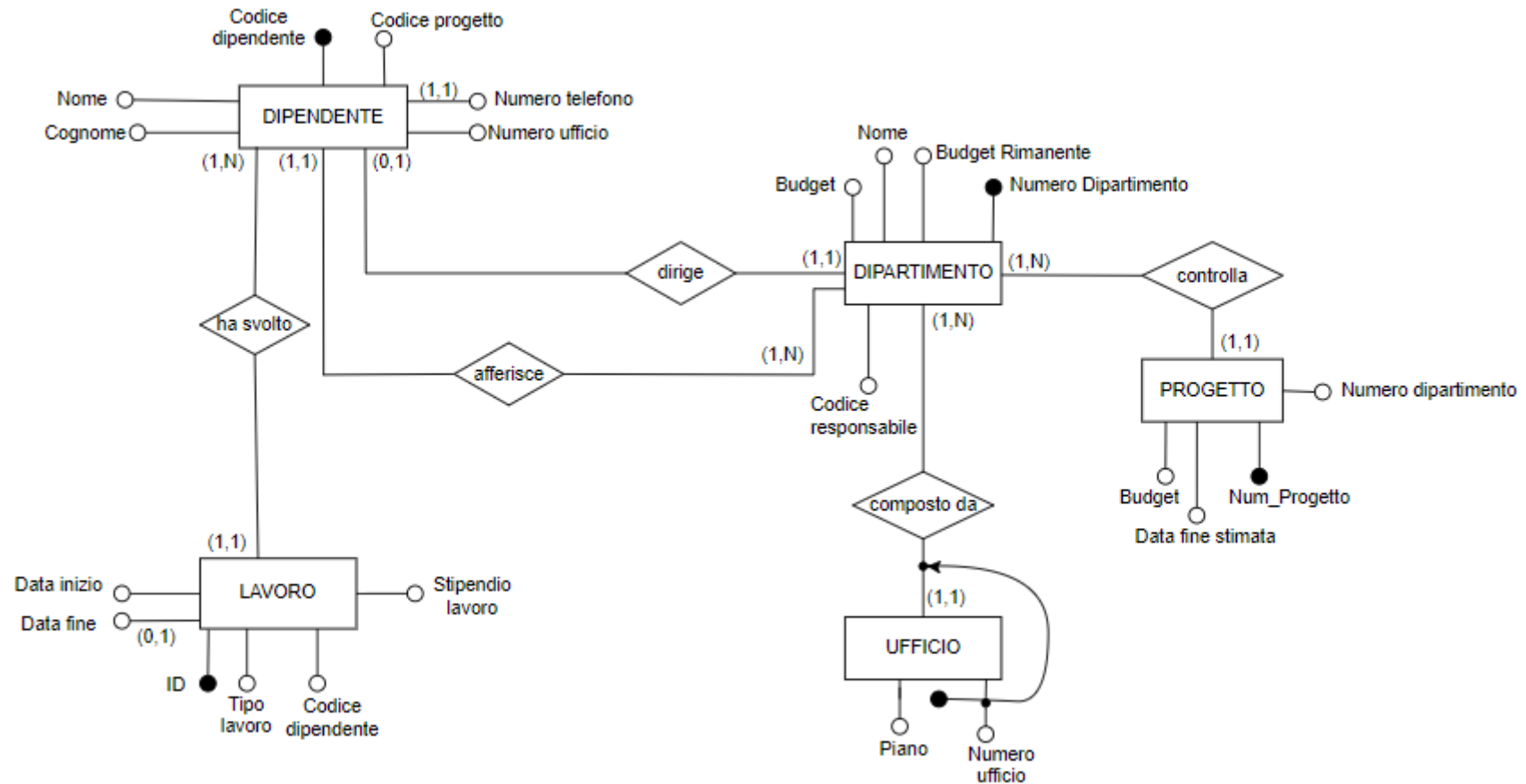
L'eliminazione delle generalizzazioni prevede la modifica di eventuali parti dell'E-R nelle quali sono presenti entità padri collegate ad entità figlie che hanno gli attributi in comune; nel nostro caso la distinzione tra le entità impiegato e responsabile, che sono state eliminate mantenendo il collegamento diretto costituito dalle rispettive relazioni all'entità dipartimento.

4.3 Scelta delle chiavi primarie

La scelta ottimale delle chiavi primarie prevede il minor numero di attributi coinvolti, pur mantenendo ovviamente il vincolo di univocità; le scelte effettuate durante la prima versione del diagramma E-R non sono state cambiate e vengono riportate di seguito tutte le chiavi primarie:

Entità	Chiave	Spiegazione
Dipendente	codice_dipendente	Numero crescente univoco che segue il progressivo inserimento dei dipendenti nel database
Dipartimento	numero_dipartimento	Numero univoco
Progetto	numero_progetto	Numero crescente univoco che segue il progressivo inserimento dei progetti nel database
Lavoro	id	Numero crescente univoco, la scelta deriva dalla natura dell'entità "lavoro", che rappresenta un impiego svolto (o in svolgimento) da parte di un impiegato, non rappresentabile univocamente da altri singoli attributi
Ufficio	numero_ufficio + numero_dipartimento	Coppia di numeri univoci crescenti; la coppia è utile per identificare uffici con lo stesso numero ma appartenenti a dipartimenti diversi

4.4 Schema E-R dopo la progettazione logica



4.5 Traduzione da schema E-R a logico

Relazione “dirige”

La relazione “dirige” indica il responsabile di un dipartimento, che è anche un dipendente dello stesso dipartimento. Tale relazione verrà indicata con un campo “responsabile” nella tabella Dipartimento che punterà ad un elemento della tabella Dipendente.

Relazione “afferisce”

La relazione “afferisce” indica un impiegato di un'azienda. Dato che un impiegato è anche un dipendente si elimina l'entità impiegato per lasciare solo l'entità più generica Dipendente.

4.6 Schema logico

DIPENDENTE(codice_dipendente, nome, cognome, telefono, *dipartimento*, *progetto*, *lavoro*, *ufficio*)

UNIQUE(telefono)

DIPENDENTE(dipartimento) → DIPARTIMENTO(numero_dipartimento)

DIPENDENTE(progetto) → PROGETTO(numero_progetto)

DIPENDENTE(lavoro) → LAVORO(id)

DIPENDENTE(ufficio) → UFFICIO(numero_ufficio)

LAVORO(id, data_inizio, data_fine, tipo_lavoro, stipendio_lavoro, *impiegato*)

LAVORO(impiegato) → DIPENDENTE(codice_dipendente)

DIPARTIMENTO(numero dipartimento, nome, budget, budget_rimanente, *responsabile*)

DIPARTIMENTO(responsabile) → DIPENDENTE(codice_dipendente)

UFFICIO(numero ufficio, *dipartimento*, piano)

UFFICIO(numero_dipartimento) → DIPARTIMENTO(numero_dipartimento)

PROGETTO(numero progetto, data_fine_stimata, budget, *dipartimento*)

PROGETTO(dipartimento) → DIPARTIMENTO(numero_dipartimento)

N.B. gli attributi sottolineati e in grassetto indicano i componenti della **chiave primaria**, mentre quelli in blu e corsivo identificano le *chiavi esterne*.

5 PROGETTAZIONE FISICA

Nella fase di progettazione fisica l'obiettivo è creare il database effettivo tramite codice: è stato utilizzato il linguaggio SQL sull'ambiente di sviluppo PostgreSQL con il supporto grafico di pgAdmin 4.

5.1 Domini

Telefono:

Per i numeri di telefono è stato creato un dominio caratterizzato da un dato di tipo testo con lunghezza 10 caratteri.

```
//Check correttezza numero di telefono:
CREATE DOMAIN public.telefono
(
    AS text;
)

ALTER DOMAIN public.telefono
(
    ADD CONSTRAINT tel_check CHECK (VALUE ~ '^\\d{10}$'::text);
)
```

5.2 Creazione tabelle

Di seguito si è proseguito con la creazione delle tabelle e relativi vincoli.

```
//Creazione tabella dipendente
CREATE TABLE IF NOT EXISTS public.dipendente
(
    codice_dipendente integer NOT NULL,
    nome text NOT NULL,
    cognome text NOT NULL,
    dipartimento integer NOT NULL,
    progetto integer NOT NULL,
    lavoro integer NOT NULL,
    ufficio integer NOT NULL,
    telefono telefono NOT NULL,
    CONSTRAINT dipendente_pkey PRIMARY KEY (codice_dipendente),
    CONSTRAINT dipendente_telefono_key UNIQUE (telefono),
    CONSTRAINT dipartimento_fk FOREIGN KEY (dipartimento)
```

```

        REFERENCES public.dipartimento (numero_dipartimento) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT lavoro_fk FOREIGN KEY (lavoro)
        REFERENCES public.lavoro (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT progetto_fk FOREIGN KEY (progetto)
        REFERENCES public.progetto (numero_progetto) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT ufficio_fk FOREIGN KEY (ufficio,dipartimento)
        REFERENCES public.ufficio (numero_ufficio, dipartimento) MATCH
SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE
)

// Creazione tabella lavoro
CREATE TABLE IF NOT EXISTS public.lavoro
(
    id integer NOT NULL,
    data_inizio date NOT NULL,
    data_fine date,
    tipo_lavoro text NOT NULL,
    stipendio_lavoro integer NOT NULL,
    impiegato integer NOT NULL,
    CONSTRAINT lavoro_pkey PRIMARY KEY (id),
    CONSTRAINT impiegato_fk FOREIGN KEY (impiegato)
        REFERENCES public.dipendente (codice_dipendente) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

// Creazione tabella progetto
CREATE TABLE IF NOT EXISTS public.progetto
(
    numero_progetto integer NOT NULL,
    data_fine_stimata date NOT NULL,
    budget integer NOT NULL,
    dipartimento integer NOT NULL,
    CONSTRAINT progetto_pkey PRIMARY KEY (numero_progetto),
    CONSTRAINT dipartimento_fk FOREIGN KEY (dipartimento)
        REFERENCES public.dipartimento (numero_dipartimento) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

```

```

)

// Creazione tabella dipartimento
CREATE TABLE IF NOT EXISTS public.dipartimento
(
    numero_dipartimento integer NOT NULL,
    nome text NOT NULL,
    budget integer NOT NULL,
    budget_rimanente integer NOT NULL,
    responsabile integer NOT NULL,
    CONSTRAINT dipartimento_pkey PRIMARY KEY (numero_dipartimento),
    CONSTRAINT responsabile_fk FOREIGN KEY (responsabile)
        REFERENCES public.dipendente (codice_dipendente) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

// Creazione tabella ufficio
CREATE TABLE IF NOT EXISTS public.ufficio
(
    numero_ufficio integer NOT NULL,
    dipartimento integer NOT NULL,
    piano integer NOT NULL,
    CONSTRAINT ufficio_pkey PRIMARY KEY (numero_ufficio, dipartimento),
    CONSTRAINT dipartimento_fk FOREIGN KEY (dipartimento)
        REFERENCES public.dipartimento (numero_dipartimento) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

```

6 IMPLEMENTAZIONE

6.1 Popolamento del DataBase

Per popolare questi dati si è deciso di modellare un'azienda produttrice di apparecchi elettronici, generando cinque tabelle tramite codice scritto in Java nell'ambiente di sviluppo Processing.

La prima parte del codice descrive le variabili globali per la generazione dei dati:

```
int NUMDIPE = 720;
int NUMLAV = 1160;
int NUMPROG = 162;
int NUMUFF = 36;

String[] dipartimenti = {"Produzione", "Testing", "Logistica", "Amministrazione",
    "Commercio", "Relazioni pubbliche", "Legale",
    "Risorse umane", "Ricerca e sviluppo"};

int[] personePerDipartimenti = {327, 76, 92, 58, 54, 23, 27, 31, 32}; // Totale: 720
int[] progettiPerDipartimenti = {41, 29, 27, 20, 16, 11, 7, 5, 6}; // Totale: 1160
int[] budgetPerDipartimenti = {12140000, 650000, 2340000, 220000, 1340000, 360000, 210000,
    190000, 1820000};
int[] budgetRimPerDipartimenti = {1420000, 130000, 230000, 10000, 80000, 110000, 40000,
    30000, 70000};
int[] ufficiPerDipartimenti = {9, 4, 3, 5, 4, 3, 4, 2, 2}; // Totale -> 36
// Gli uffici si trovano tra i piani PIANO e PIANO-1
int[] pianiPerDipartimenti = {1, 2, 2, 5, 4, 5, 5, 3, 2};

String[][] lavoriPerDipartimenti = {{"Operaio", "Addetto ai macchinari",
    "Manutentore", "Programmatore", "Ingegnere"},
    {"Operaio", "Tecnico", "Supervisore", "Specialista"},
    {"Magazziniere", "Autista", "Direttore logistica"},
    {"Supervisore", "Manager", "Amministratore",
    "Gestore", "Direttore"},
    {"Consulente", "Ragioniere", "Analista",
    "Commercialista"},
    {"Mediatore", "Organizzatore", "Rappresentante"},
    {"Segretario", "Consulente avvocato", "Avvocato"},
    {"Segretario", "Supervisore",
    "Amministratore del personale"},
    {"Ricercatore", "Analista", "Ingegnere ricercatore"}};
```

Tra queste è già presente un insieme di dipartimenti stabiliti in partenza e le loro corrispondenti informazioni tra cui nome, numero dipendenti, budget e tipi di lavori presenti nel dipartimento.

Queste informazioni andranno a costituire la struttura fondamentale del nostro dataset, mentre per le informazioni più specifiche e ripetitive si sono studiate altre vie di generazione:

- Per i nomi dei dipendenti sono state trovate due liste di nomi e cognomi italiani già esistenti e si sono create coppie casuali tra di esse;
- Per i numeri di telefono sono stati generati dei numeri pseudo casuali;
- Per gli id dei dipendenti e dei lavori sono stati scelti numeri interi in ordine crescente (con l'assenza di alcuni di questi, per tenere in considerazione i dipendenti che non lavorano più in azienda)
- Per gli stipendi sono stati generati dei valori casuali tra dei range prestabiliti (ritenuti plausibili per il tipo di lavoro corrispondente);
- Per i budget dei progetti sono stati generati dei valori in modo casuale, con il vincolo che la loro somma non superi il budget totale del dipartimento di cui fanno parte;

Tutti i dati casuali sono poi stati aggregati rispettando i CONSTRAINTS del progetto.

Ad esempio, la tabella 'dipendente' è stata creata e popolata nel seguente modo:

```
// Genera tabelle
Table DIPENDENTE = new Table();
DIPENDENTE.addColumn("codice_dipendente");
DIPENDENTE.addColumn("nome");
DIPENDENTE.addColumn("cognome");
DIPENDENTE.addColumn("telefono");
DIPENDENTE.addColumn("dipartimento");
DIPENDENTE.addColumn("progetto");
DIPENDENTE.addColumn("lavoro");
DIPENDENTE.addColumn("ufficio");

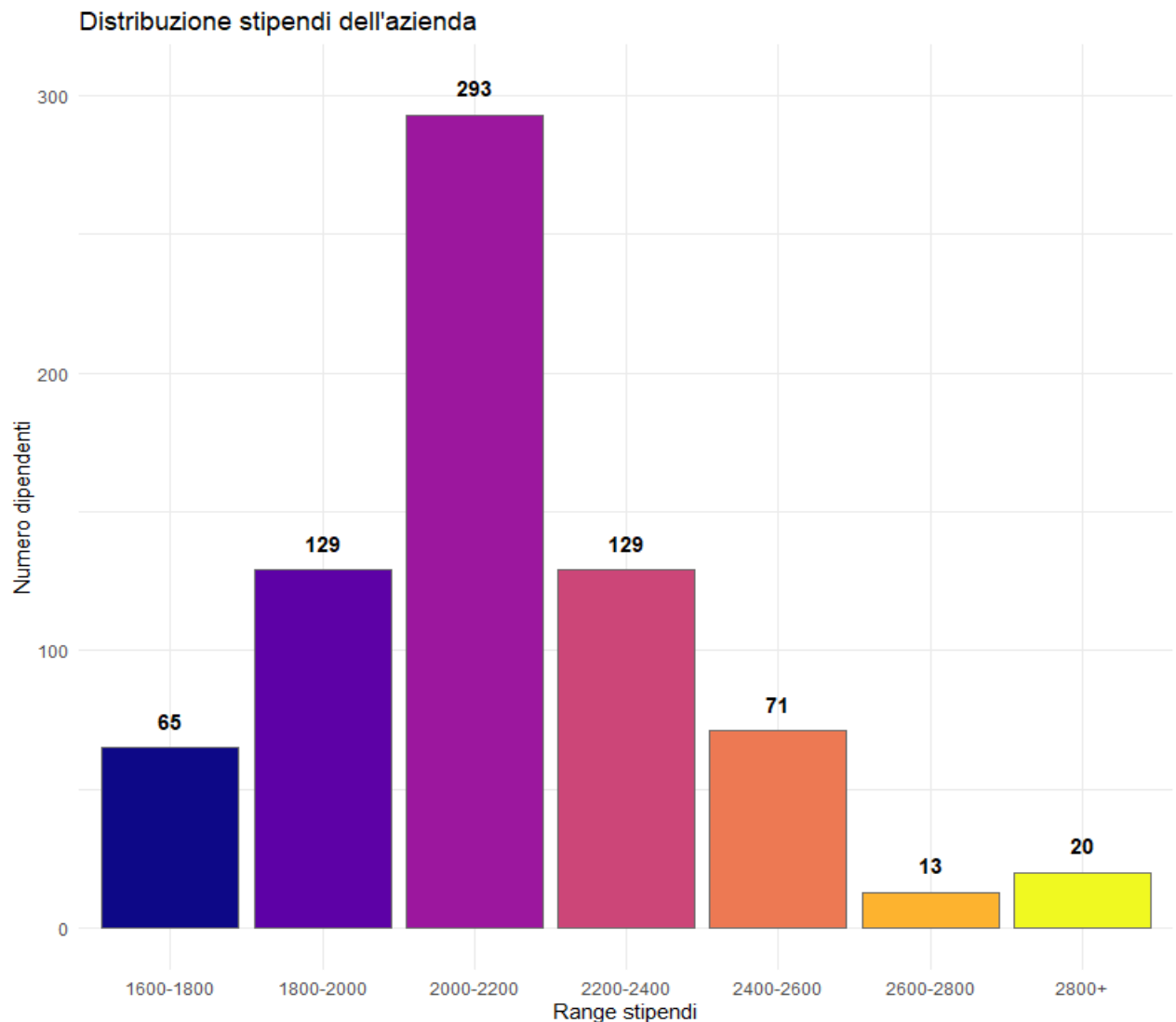
for (int i = 0; i < dipendenti.length; ++i) {
    TableRow newRow = DIPENDENTE.addRow();
    newRow.setString("codice_dipendente", dipendenti[i][2]);
    newRow.setString("nome", "" + dipendenti[i][0] + "");
    newRow.setString("cognome", "" + dipendenti[i][1] + "");
    newRow.setString("telefono", "" + dipendenti[i][3] + "");
    newRow.setString("dipartimento", dipendenti[i][5]);
    newRow.setString("progetto", dipendenti[i][7]);
    newRow.setString("lavoro", dipendenti[i][6]);
    newRow.setString("ufficio", dipendenti[i][8]);
}
```

Infine, i dati sono stati salvati nei rispettivi file di estensione .csv che poi saranno importati nel database:

```
if (SAVING) {  
    saveTable(DIPENDENTE, "dipendente.csv");  
    saveTable(DIPARTIMENTO, "dipartimento.csv");  
    saveTable(LAVORO, "lavoro.csv");  
    saveTable(PROGETTO, "progetto.csv");  
    saveTable(UFFICIO, "ufficio.csv");  
}
```


7 ANALISI DEI DATI IN R

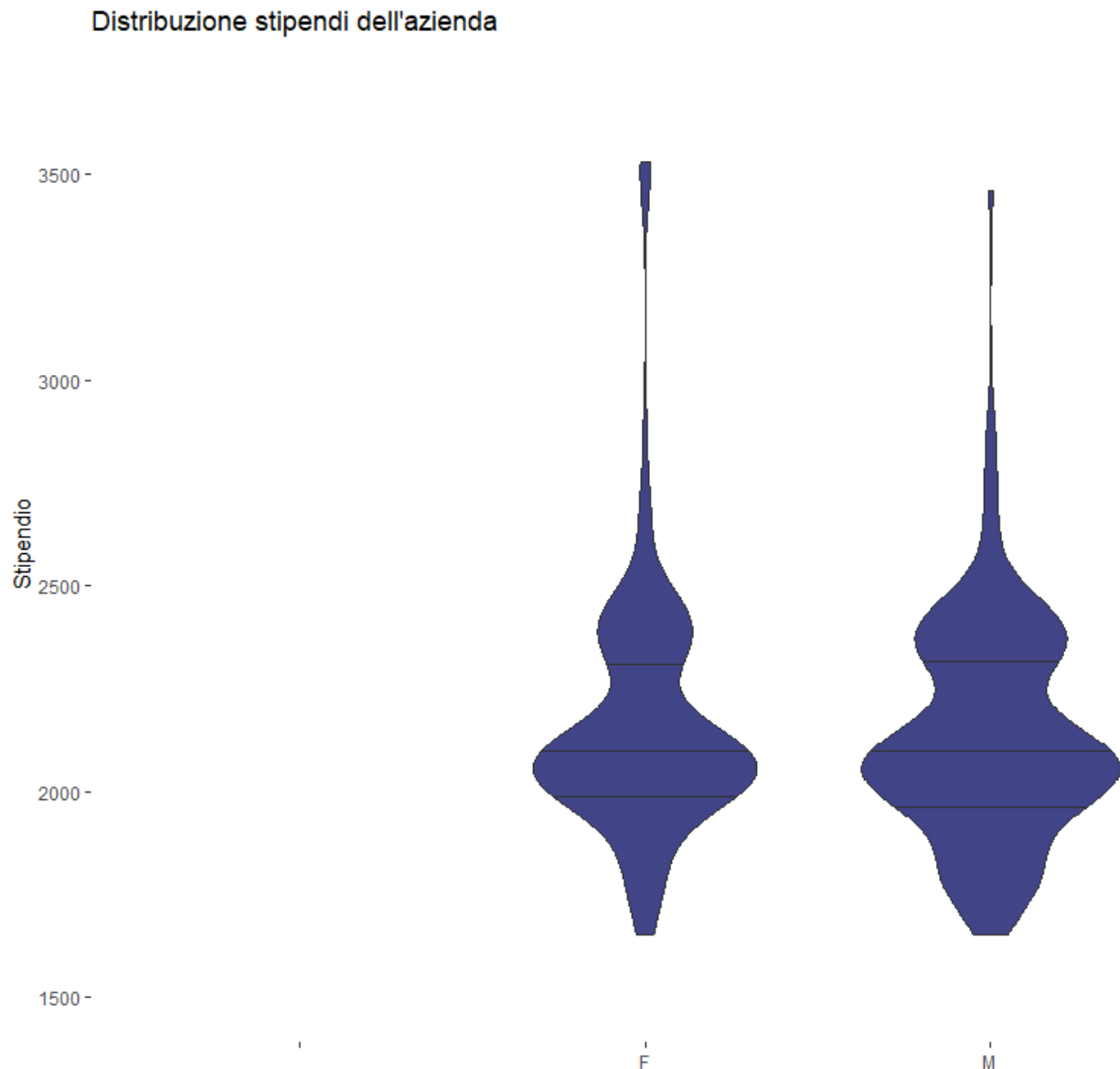
La parte finale della relazione è un'analisi dei dati su cui si basa il progetto:



Questo è un istogramma che rappresenta la distribuzione degli stipendi lordi dell'azienda (divisi per range).

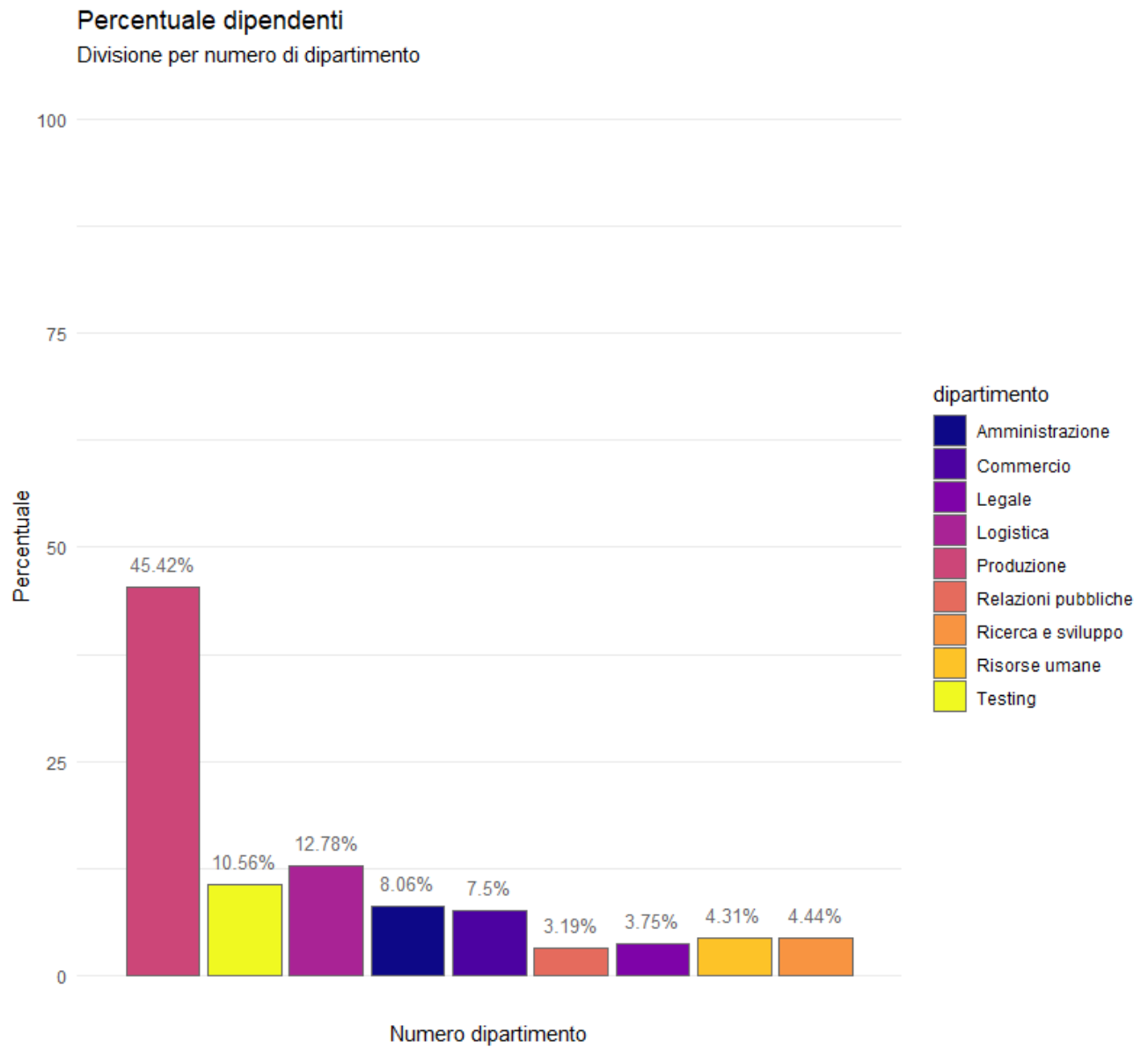
Gli stipendi più comuni sono quelli compresi tra €2000 e €2200; si può notare che la distribuzione degli stipendi ha una forma a campana, rendendo così più realistici i dati generati.

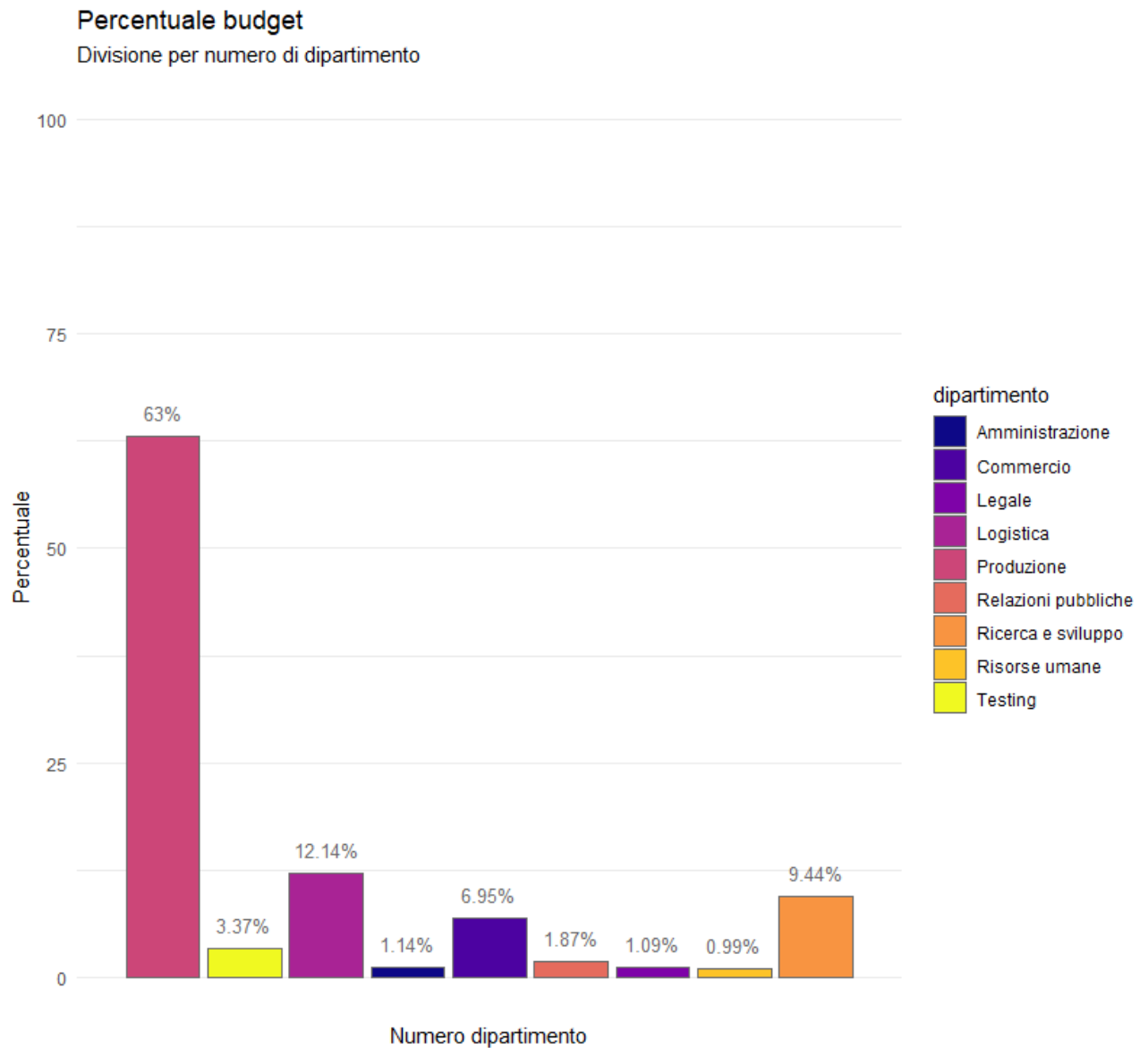
Abbiamo inoltre creato un diagramma a violino per rappresentare il medesimo dato (questa volta senza divisione per range) con l'aggiunta della divisione maschio/femmina:



Come è possibile osservare, non risultano esserci grosse differenze nella distribuzione degli stipendi tra maschi e femmine; c'è da far notare che nei dati mostrati nel grafico l'individuo con stipendio maggiore è femmina, ma è stato eliminato l'outlier con lo stipendio più alto (il proprietario) per ragioni legate alla qualità di rappresentazione dei dati.

Una serie di dati ancor più interessanti da visualizzare sono quelli relativi alla distribuzione dei dipendenti e del budget nei vari dipartimenti:





Il dipartimento che ha il maggior quantitativo di denaro a disposizione (63%) ha meno della metà dei dipendenti che ci lavorano (~45%).

Altri dipartimenti mostrano, invece, una discrepanza tra i dati, come:

- “Testing”;
- “Amministrazione”;
- “Ricerca e Sviluppo”;

Hanno invece una percentuale simile il dipartimento “Legale” e “Logistica”.