PROGETTO
BASI DI DATI

ANNO ACCADEMICO
2020/2021

Flutra Bajraktari – 142579
Camilla Burlon – 141878
Emma Curtolo – 144050
Margherita Monte – 137437

Sommario

1. TESTO DEL PROBLEMA

Si vuole realizzare una base di dati per la gestione di un insieme di informazioni circa i dipendenti, le loro competenze, i progetti a cui partecipano e i dipartimenti a cui afferiscono di una data società sulla base del seguente insieme di requisiti.

Ogni dipendente ha una matricola, che lo identifica univocamente, assegnata dalla società. Di ogni dipendente interessano il nome e il cognome, la data di nascita e la data di assunzione. Se un dipendente è coniugato con un altro dipendente della stessa società, interessano la data del matrimonio e il coniuge. Ogni dipendente ha una qualifica (ad esempio, amministrativo, commerciale, programmatore, analista, progettista, ecc.).

Dei dipendenti in possesso di un titolo di studio superiore (laurea ed, eventualmente, dottorato di ricerca) interessa conoscere la classe di laurea (informatica, matematica, fisica, ..) e la data di conseguimento della laurea, ed, eventualmente, la classe di dottorato (informatica, matematica, fisica, ..) e la data di conseguimento del dottorato. Per semplicità, si assuma che:

i. Ogni dipendente possieda al più una laurea,
ii. ogni dipendente possieda al più dottorato e
iii. per conseguire il titolo di dottore di ricerca, un dipendente debba possedere una laurea (non vale, ovviamente, il viceversa: non tutti i laureati possiedono anche un dottorato).

L'attività dell'azienda è strutturata in progetti. Ogni progetto è identificato da un codice aziendale ed è caratterizzato da un budget e da una durata espressa in mesi. Ogni progetto coinvolge uno o più dipendenti.

Ogni dipendente possiede un certo numero di competenze (una o più) e lavora a uno o più progetti. In ogni progetto a cui lavora, un dipendente usa una o più delle sue competenze, non necessariamente tutte. Si vuole tener traccia delle competenze usate da un dipendente in ogni progetto al quale partecipa.

La società è organizzata in dipartimenti identificati da un nome e caratterizzati da un recapito telefonico e da un indirizzo di posta elettronica. Ogni dipendente afferisce ad un solo dipartimento. Ogni dipartimento si approvvigiona presso vari fornitori e un fornitore può rifornire vari dipartimenti. Di ogni fornitore interessano il nome e l'indirizzo. Interessano, inoltre, la data e il fornitore dell'ultimo acquisto fatto da un dipartimento.

2. RACCOLTA ED ANALISI DEI REQUISITI

2.1 Glossario dei termini

Termine	Descrizione	Sinonimo	Collegamenti
Dipendente	Lavora nella società ed è identificato da una matricola assegnata dalla società stessa.	Coniuge	Dipartimento, progetto, competenze
Dipartimento	Società organizzata in dipartimenti, identificati da un nome.	-	Dipendente, fornitore
Competenze	Abilità di un determinato dipendente, identificato dal tipo.	-	Dipendente, progetto
Progetto	L'attività dell'azienda è strutturata in progetti, identificati da un codice aziendale.	-	Competenze, dipendente
Fornitore	Rifornisce i vari dipartimenti, identificato da un nome.	-	Dipartimento,

2.2 Riscrittura e strutturazione dei requisiti

Frasi di natura generale

Si vuole realizzare una base di dati per la gestione di un insieme di informazioni circa i dipendenti, le loro competenze, i progetti a cui partecipano e i dipartimenti a cui afferiscono di una data società sulla base del seguente insieme di requisiti.

Frasi relative ai dipendenti

Ogni dipendente ha una matricola, che lo identifica univocamente, assegnata dalla società. Di ogni dipendente interessano il nome e il cognome, la data di nascita e la data di assunzione. Se un dipendente è coniugato con un altro dipendente della stessa società, interessano la data del matrimonio e il nome del coniuge.
Ogni dipendente ha una qualifica.
Dei dipendenti in possesso di un titolo di studio superiore interessa conoscere la classe di laurea (informatica, matematica, fisica, ..) e la data di conseguimento della laurea, eventualmente, la classe di dottorato e la data di conseguimento del dottorato.
Per semplicità, si assuma che ogni dipendente possieda al più una laurea e al più un dottorato (per conseguire il titolo di dottore di ricerca, un dipendente deve possedere una laurea).

Frasi relative ai dipartimenti

La società è organizzata in dipartimenti identificati da un nome e caratterizzati da un recapito telefonico e da un indirizzo di posta elettronica.
Ogni dipendente afferisce ad un solo dipartimento.
Ogni dipartimento si approvvigiona presso vari fornitori e un fornitore può rifornire vari dipartimenti.

Frasi relative alle competenze

Ogni dipendente possiede un certo numero di competenze (una o più) e lavora a uno o più progetti. In ogni progetto a cui lavora, un dipendente usa una o più delle sue competenze, non necessariamente tutte. Si vuole tener traccia delle competenze usate da un dipendente in ogni progetto al quale partecipa.

Frasi relative ai progetti

L'attività dell'azienda è strutturata in progetti. Ogni progetto è identificato da un codice aziendale ed è caratterizzato da un budget e da una durata espressa in mesi.
Ogni progetto coinvolge uno o più dipendenti.

Frasi relative ai fornitori

Di ogni fornitore interessano il nome e l'indirizzo.
Ogni dipartimento si approvvigiona presso vari fornitori e un fornitore può rifornire vari dipartimenti.

3. PROGETTAZIONE CONCETTUALE

La progettazione dello schema Entità-Relazione è avvenuta in quattro fasi.

3.1 Prima fase

Nella prima fase sono state identificate le entità più rilevanti e successivamente sono state rappresentate in uno schema *scheletro* che è servito come "base" per lo schema E-R.

Il processo parte con l'entità DIPENDENTE che è il punto chiave della società, infatti esso lavora a diversi PROGETTI e afferisce a un solo DIPARTIMENTO.

Ogni DIPARTIMENTO è approvvigionato da uno o più FORNITORI.

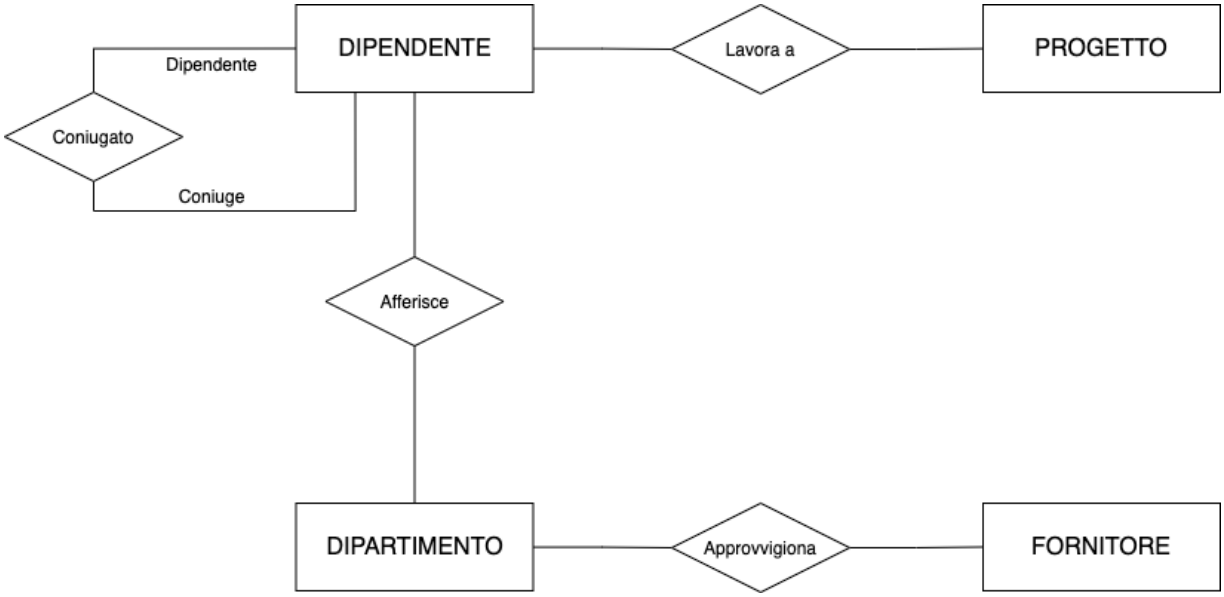

Figura 1 - Prima fase

Le successive fasi della progettazione dello schema E-R consistono nel vedere nel dettaglio i requisiti con riferimento ai concetti espressi nello schema scheletro.

3.2 Seconda fase

Nella seconda fase è stata analizzata l'entità DIPENDENTE e le varie entità con cui è in relazione. L'entità DIPENDENTE è caratterizzata dai seguenti attributi: *Matricola* (che lo identifica in modo univoco), *Nome, Cognome, Data Di Nascita, Data Di Assunzione, Qualifica,* dall'attributo composto *Titolo Di Studio* (*Laurea, Data Di Laurea, Dottorato e Data del dottorato*), i quali sono opzionali, e dall'attributo multivalore *Competenza*.
Si richiede inoltre di specificare se esiste un eventuale matrimonio tra due dipendenti. Il problema viene gestito attraverso la relazione ricorsiva CONIUGATO che registra la data del matrimonio utilizzando l'attributo *Data Di Matrimonio*.

Successivamente, è stata modellata l'entità PROGETTO, identificata univocamente dall'attributo *Codice Aziendale* e caratterizzata da *Budget* e *Durata*.

La relazione LAVORA A, che collega DIPENDENTE e PROGETTO, è necessaria per associare ogni dipendente a tutti i progetti a cui ha lavorato con le sue relative competenze utilizzate attraverso l'attributo *Competenza*.

Inoltre, ciascun DIPENDENTE afferisce ad un unico DIPARTIMENTO identificato dal *Nome* e contrassegnato dal *Telefono* e dalla *E-mail*.

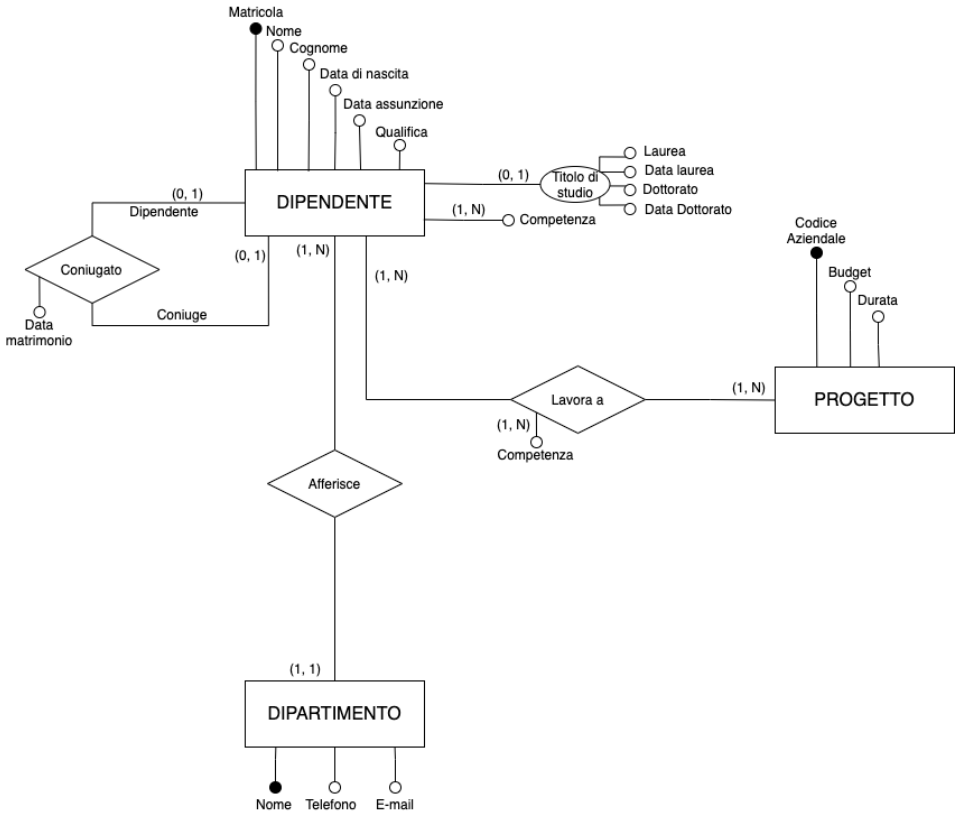

Figura 2 - Seconda fase

3.3 Terza fase

Nella terza fase è stata modellata l'entità FORNITORE, è caratterizzato dai seguenti attributi: *Nome* e *Indirizzo*. Nel testo non è specificato quale sia l'attributo che lo identifica univocamente, quindi abbiamo ritenuto opportuno adottare una chiave composta formata da entrambi gli attributi.

Il testo richiede di memorizzare la data e il fornitore dell'ultimo acquisto; tutto ciò è stato eseguito inserendo un attributo composto *Ultimo Acquisto*(*Data* e *Fornitore*) nella relazione APPROVVIGIONA, che collega DIPARTIMENTO e FORNITORE.

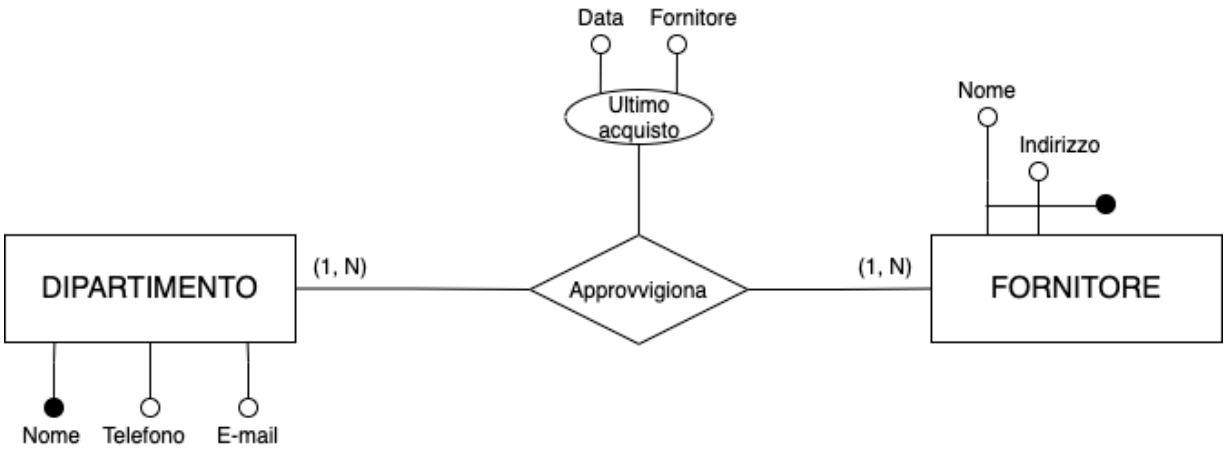

Figura 3 - Terza fase

3.4 Quarta fase

Nella quarta e ultima fase sono state unite le due scomposizioni principali (*Figura 2 e 3*) per ottenere lo schema E-R completo.

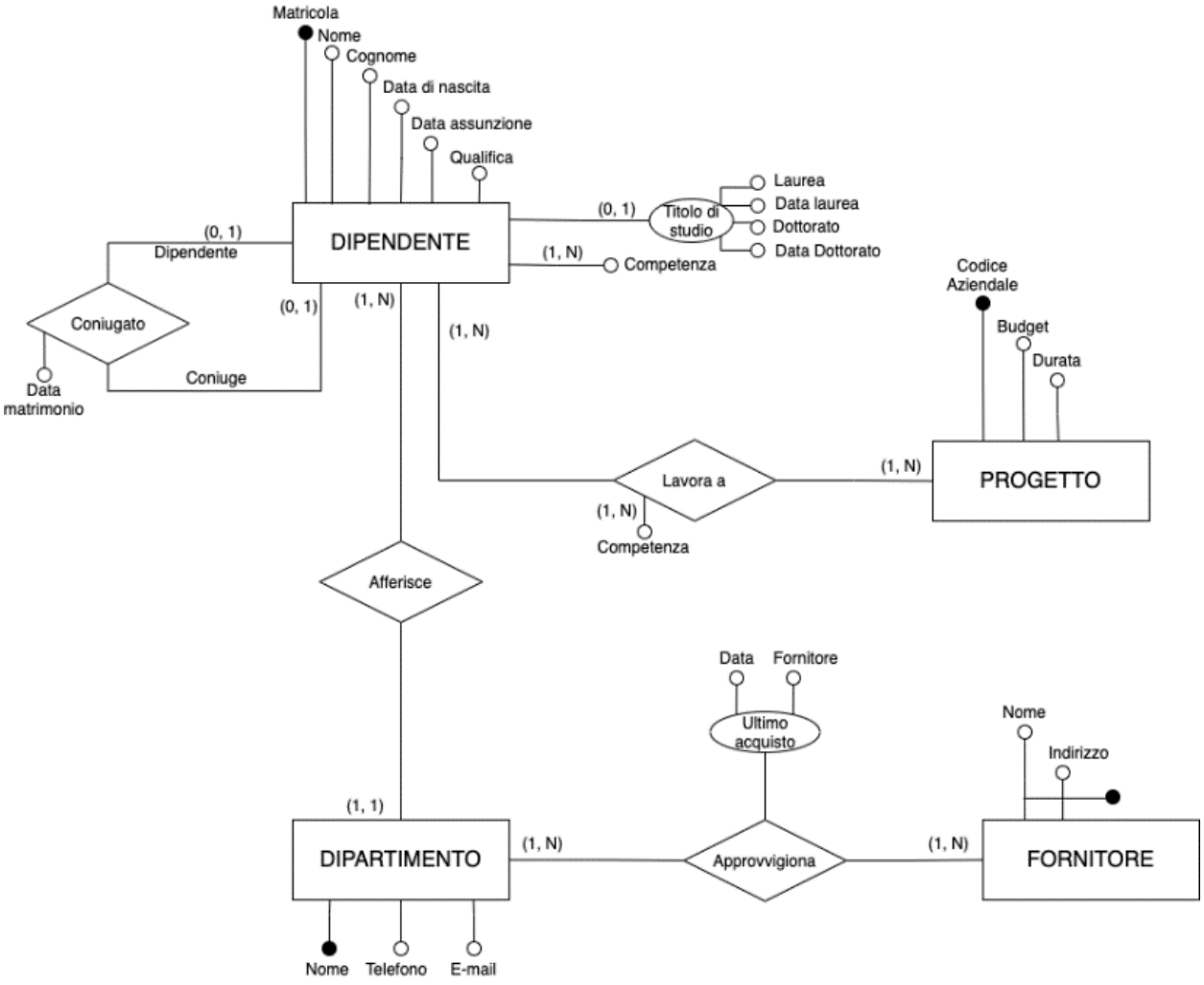

Figura 4 - Quarta fase, schema ER completo

4. PROGETTAZIONE LOGICA

4.1 Ristrutturazione del modello E-R

È utile semplificare strutture come attributi composti e attributi multivalore perché questo renderà più facile la traduzione dal modello Entità-Relazione al modello Relazionale.

4.1.1 Semplificazione dei concetti

Per semplificare l'attributo multivalore *Competenza* in DIPENDENTE è stata modellata una relazione HA COMPETENZE tra DIPENDENTE e la nuova entità COMPETENZA. Quest'ultima ha come attributo chiave primaria *Tipo*.

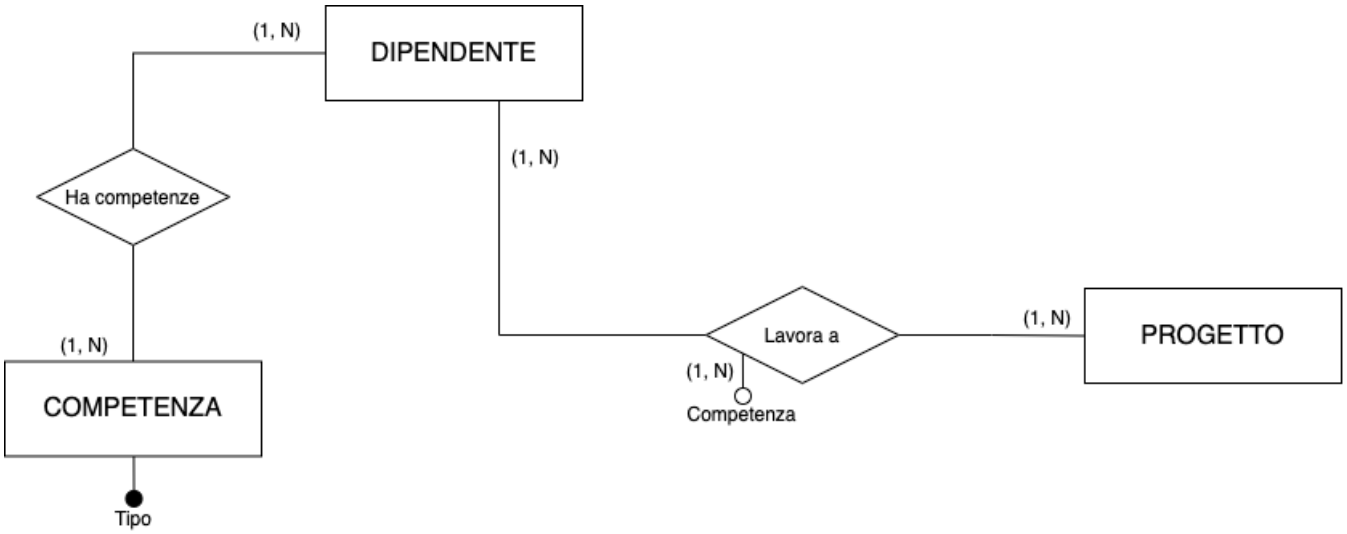

Figura 5 - Semplificazione attributo multivalore

In seguito sono stati semplificati gli attributi composti, poiché il modello relazionale non permette ad un attributo di avere più di un valore.

Nel caso dell'attributo *Titolo Di Studio* in DIPENDENTE è stato scelto di creare quattro nuovi attributi: *Laurea, Data Di Laurea, Dottorato e Data del dottorato* che, come detto in precedenza (Sezione 3.2), rimangono opzionali.

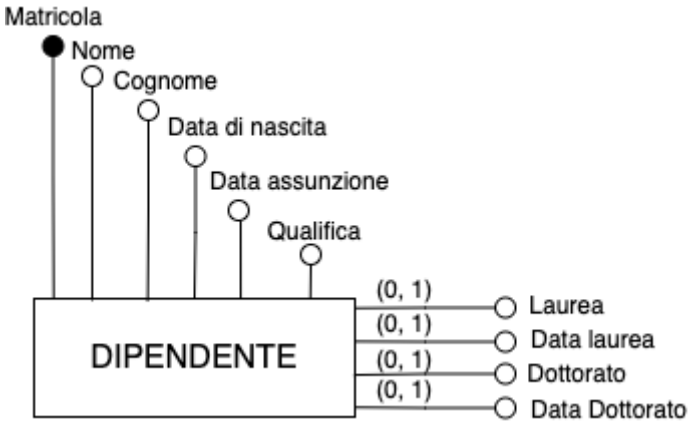

Figura 6 - Semplificazione attributo composto

Anche nel caso dell'attributo *Ultimo Acquisto* in APPROVVIGIONA è stato scelto di scomporlo: aggiungendo un attributo *Data* in APPROVVIGIONA che va a memorizzare la data di ogni approvvigionamento, mentre, i dati relativi all'ultimo acquisto (*Data Ultimo Acquisto, Ultimo Fornitore*) essendo specifici per ogni dipartimento è stato ritenuto più pratico spostarli nell'entità DIPARTIMENTO.

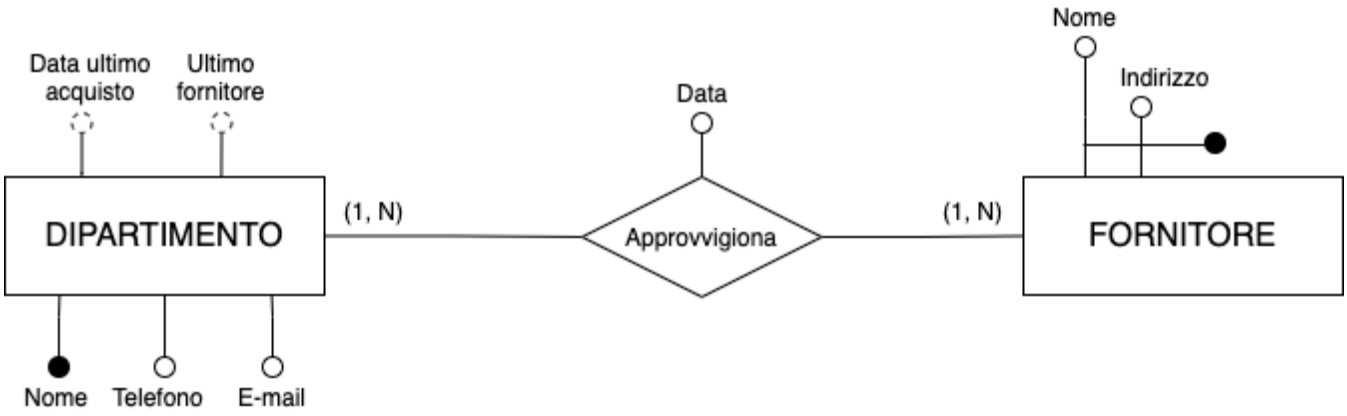

Figura 7 - Semplificazione e modifica attributo composto

4.2 Analisi delle ridondanze

L' analisi delle ridondanze è un processo necessario per verificare l'effettiva utilità degli attributi derivati presenti nella base di dati; questi introducono infatti informazioni doppie, occupando memoria, ma soprattutto possono portare a pericolosi conflitti. Il confronto che si osserva in questa analisi consente di capire se le informazioni che gli attributi derivati forniscono sarebbero dispendiose da ottenere in altro modo.

4.3 Analisi degli attributi derivati

L'attributo *Data Ultimo Acquisto* dell'entità DIPARTIMENTO è derivato in quanto può essere ricavato utilizzando la funzione MAX sull'attributo *Data* presente in APPROVVIGIONA così da estrarre quella più recente per ogni dipartimento.

L'attributo *Ultimo Fornitore* dell'entità DIPARTIMENTO è derivato in quanto può essere ricavato prendendo il nome e l'indirizzo del FORNITORE che approvvigiona un determinato dipartimento nella data estratta.

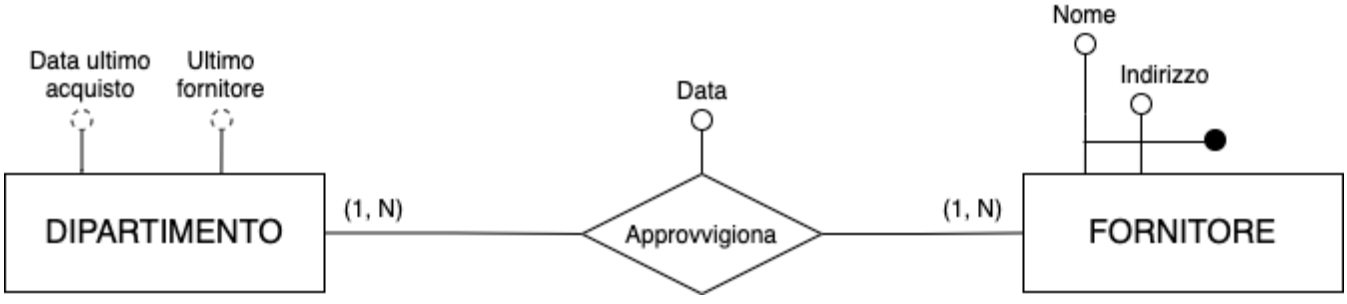

Figura 8 - Relazione con attributo ridondante

Si suppone di avere i seguenti volumi (Tabella 1) e le seguenti operazioni (Tabella 2) :

Concetto	Tipo	Volume
DIPARTIMENTO	E	12 (dipartimenti)
FORNITORE	E	30 (fornitori)
APPROVVIGIONA	R	100000 (acquisti)

Tabella 1: Tabella dei volumi

Operazione	Tipo	Frequenza
Acquisto	I	250/giorno
Stampare la data dell'ultimo acquisto	I	Dopo ogni acquisto
Stampare informazioni sull'ultimo fornitore	I	Dopo ogni acquisto

Tabella 2: Tabella delle operazioni

Per ogni operazione ipotizzata è stato calcolato il numero di accessi totale, in scrittura e in lettura, alle relazioni o entità coinvolte.

Tabella degli accessi, caso con ridondanza:

Concetto	Costrutto	Accesso	Tipo
Acquisto (Op 1)			
DIPARTIMENTO	E	1	R
FORNITORE	E	1	R
APPROVVIGIONA	R	1	W
Stampare la data dell'ultimo acquisto (Op 2)			
DIPARTIMENTO	E	1	R
Stampare informazioni sull'ultimo fornitore (Op 3)			
DIPARTIMENTO	E	1	R

Tabella 3: Tabella degli accessi caso con ridondanza

Considerando il costo di 1 accesso in scrittura come 2 in lettura, si calcola il numero di accessi giornalieri complessivi per le quattro operazioni :

Op1: (1R + 1R + 1W) x 250 = (1R + 1R + 2R) x 250 = 1000R

Op2: 1R x 250 = 250R

Op3: 1R x 250 = 250R

Totale = 1500R

Tabella degli accessi, caso senza ridondanza:

Concetto	Costrutto	Accesso	Tipo
Acquisto (Op 1)			
DIPARTIMENTO	E	1	R
FORNITORE	E	1	R
APPROVVIGIONA	R	1	W
Stampare la data dell'ultimo acquisto (Op 2)			
DIPARTIMENTO	E	1	R
FORNITORE	E	1	R
APPROVVIGIONA	R	8333	R
Stampare informazioni sull'ultimo fornitore (Op 3)			
FORNITORE	E	1	R
DIPARTIMENTO	E	1	R
APPROVVIGIONA	R	508333	R

Tabella 4: Tabella degli accessi caso con ridondanza

Si effettuano in media 8333 accessi in lettura alla relazione APPROVVIGIONA (numero di acquisti/numero di dipartimenti) per calcolare la data dell'ultimo acquisto dato che bisognerà accedere in lettura ogni volta per ogni acquisto fatto da ogni dipartimento per trovare la data più recente da stampare dopo ogni acquisto.

Per derivare le altre informazioni del fornitore relativo all'ultimo acquisto bisogna tenere in considerazione che andrà prima calcolata la data dell'ultimo acquisto per poi accedere in media ad almeno metà dei dati relativi agli acquisti per ricavare le informazioni del fornitore che ha approvvigionato quel dipartimento in quella determinata data.

Considerando il costo di 1 accesso in scrittura come 2 in lettura, si calcola il numero di accessi giornalieri complessivi per le quattro operazioni :

Op1: (1R + 1R + 1W) x 250 = (1R + 1R + 2R) x 250 = 1000R

Op2: (1R + 1R + 8333R) x 250 = 2083750R

Op3: (1R + 1R + 508333R) x 250 = 127083750R

Totale = 129168500R

Dall'analisi delle tabelle è risultato che nel caso con ridondanza si effettuano molti meno accessi alla base di dati rispetto alla versione senza ridondanza che invece impiega un numero molto alto di accessi visti i calcoli ripetuti, motivo per cui è conveniente mantenerla.

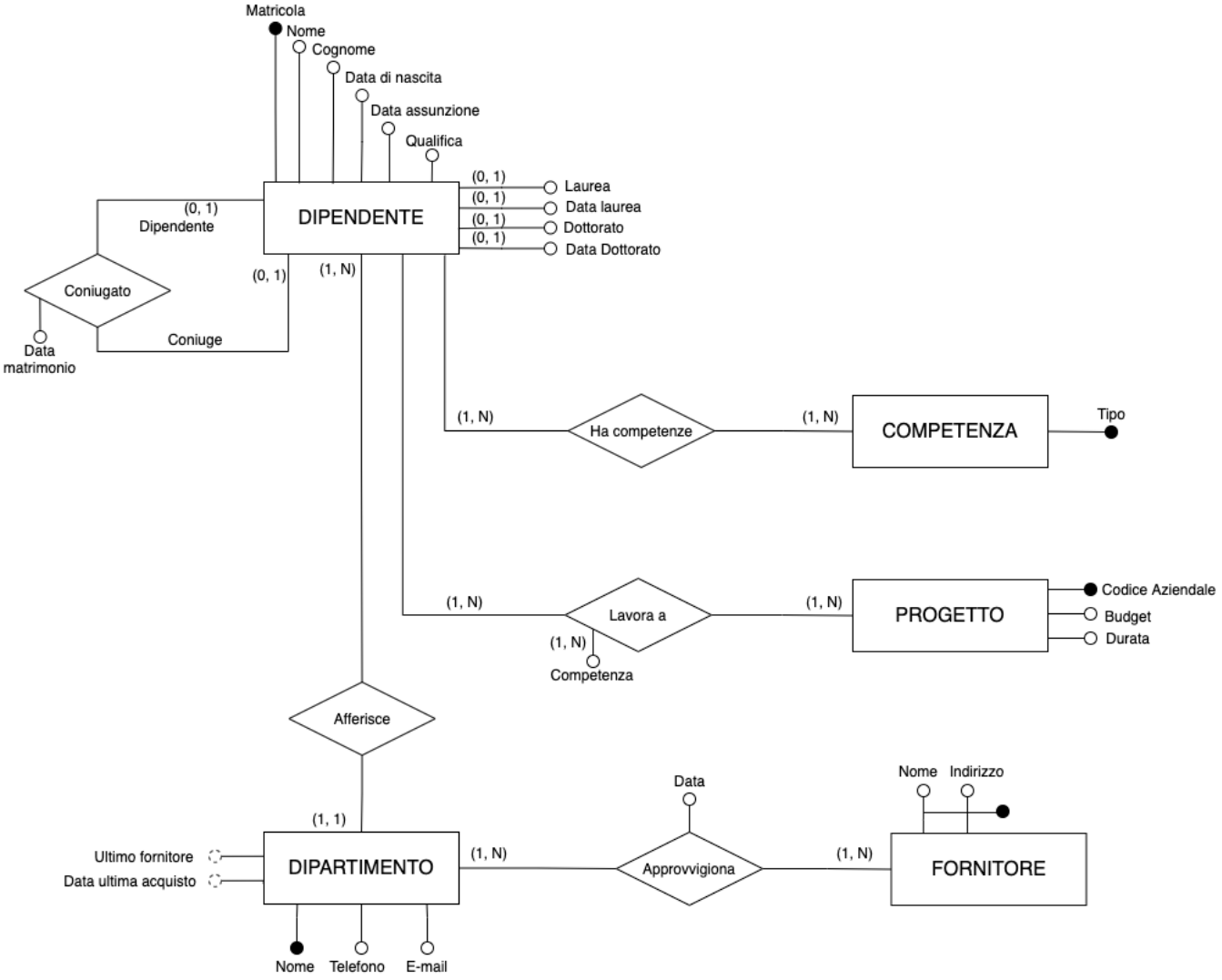

Figura 9 - schema ER finale ristrutturato

4.4 Traduzione da E-R a Relazionale

In questa fase della progettazione logica è necessario effettuare la traduzione del modello E-R in quello Relazionale. Ogni entità viene tradotta in una tabella con i relativi attributi e le chiavi, le relazioni (in base alla cardinalità) vengono tradotte o con una tabella o con un riferimento di chiave esterna.

Nella traduzione della relazione molti a molti APPROVVIGIONA è stato scelto di aggiungere una chiave primaria artificiale (*id_ordine*) la quale ci ha permesso di evitare di usare una superchiave composta da tutti e 4 gli altri attributi.

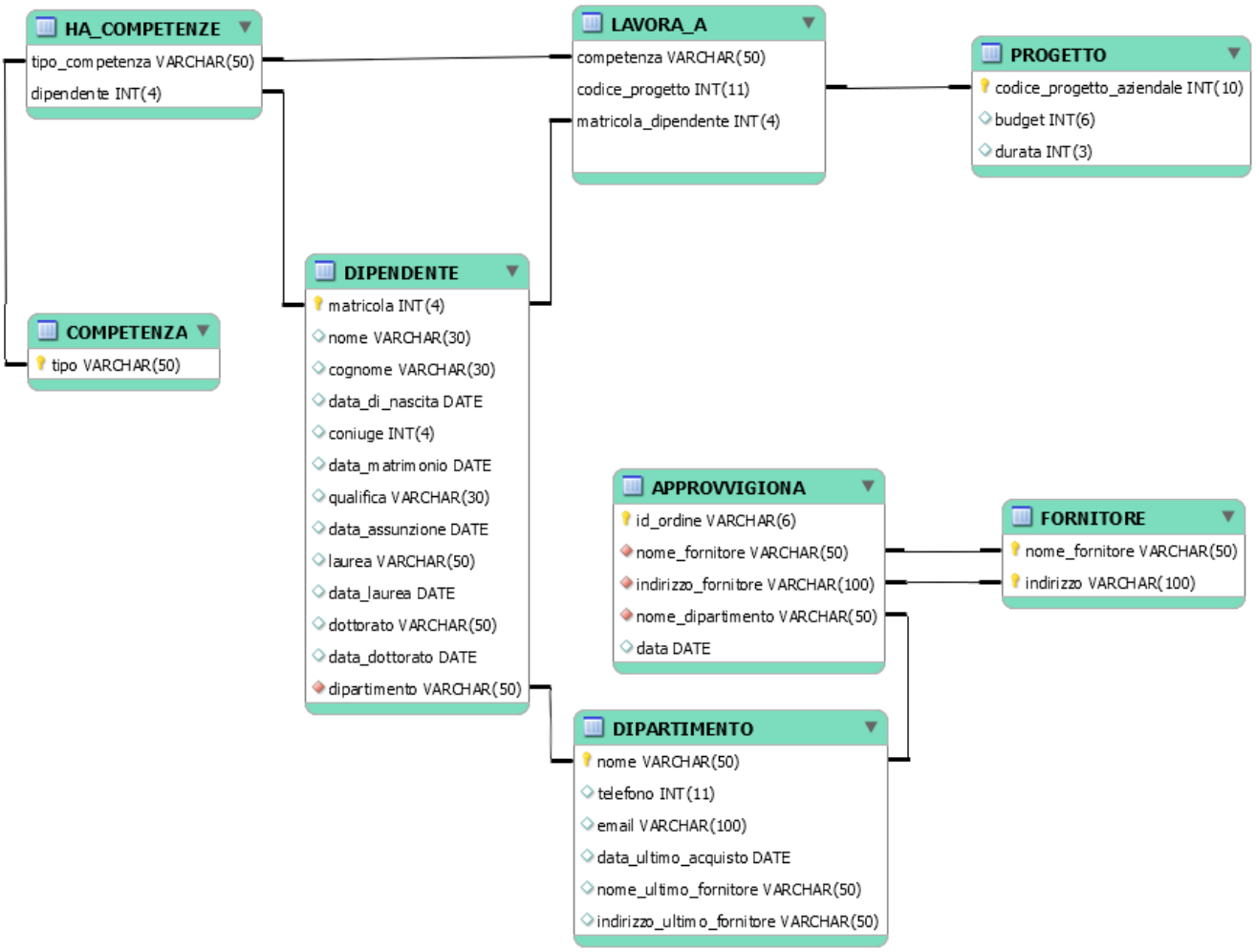

Figura 10 - schema relazionale

5. DEFINIZIONE DELLA BASE DI DATI IN SQL

Lo schema relazionale prodotto verrà utilizzato per creare la base di dati. Dovrà essere definita una tabella per ogni relazione dello schema in figura 10.

Dopodiché saranno create funzioni, constraints e trigger per garantire i vincoli di integrità.

Il primo blocco da eseguire è il seguente:

```
create database societa;
```

5.1 Definizione dei domini

Adesso è necessario definire i seguenti domini:

```
create domain dom_email as varchar check(value ~'^[a-zA-Z0-9._-]+@[a-zA-Z0-
9.-]+\.[a-zA-Z]{2,6}$');
create domain dom_tel as varchar check(value ~ '[0-9]{10}');
create domain dom_id_ordine as varchar check(value ~ '[A-Z]{2}[0-9]{4}');
create domain dom_matricola as varchar check(value ~ '[0-9]{4}');
```

5.2 Definizione delle tabelle

Procediamo ora alla creazione delle tabelle:

```
CREATE TABLE DIPENDENTE (
    matricola dom_matricola PRIMARY KEY,
    nome varchar(30) NOT NULL,
    cognome varchar(30) NOT NULL,
    data_di_nascita date NOT NULL,
    coniuge int default NULL,
    data_matrimonio date default NULL,
    qualifica varchar(30) NOT NULL,
    data_assunzione date NOT NULL,
    laurea varchar(50) default NULL,
    data_laurea date default NULL,
    dottorato varchar(50) default NULL,
    data_dottorato date default NULL,
    dipartimento varchar(50) NOT NULL
);

CREATE TABLE COMPETENZA(
    tipo varchar(50) PRIMARY KEY
);

CREATE TABLE FORNITORE (
    nome_fornitore varchar(50),
    indirizzo varchar(100),
    PRIMARY KEY (nome_fornitore, indirizzo)
);
```

```
CREATE TABLE DIPARTIMENTO(
    nome varchar(50) PRIMARY KEY,
    telefono dom_tel NOT NULL UNIQUE,
    email dom_email NOT NULL UNIQUE,
    data_ultimo_acquisto date NOT NULL,
    nome_ultimo_fornitore varchar(50) NOT NULL,
    indirizzo_ultimo_fornitore varchar(100) NOT NULL,
    FOREIGN KEY (nome_ultimo_fornitore, indirizzo_ultimo_fornitore)
    REFERENCES FORNITORE(nome_fornitore, indirizzo)
    ON UPDATE CASCADE ON DELETE CASCADE
);

ALTER TABLE DIPENDENTE
add constraint dipartimento
FOREIGN KEY(dipartimento) REFERENCES DIPARTIMENTO(nome);

CREATE TABLE PROGETTO (
    codice_progetto_aziendale int PRIMARY KEY,
    budget int NOT NULL,
    durata int NOT NULL constraint durata_valida check (durata > 0)
);

ALTER TABLE PROGETTO
add constraint budget_limiti
check (budget between 1000 and 200000);

CREATE TABLE HA_COMPETENZE(
    tipo_competenza varchar(50) NOT NULL,
    dipendente dom_matricola NOT NULL,
    PRIMARY KEY (tipo_competenza, dipendente),
    FOREIGN KEY (tipo_competenza) REFERENCES COMPETENZA(tipo) ON UPDATE
    CASCADE ON DELETE CASCADE,
    FOREIGN KEY (dipendente) REFERENCES DIPENDENTE(matricola) ON UPDATE
    CASCADE ON DELETE CASCADE
);

CREATE TABLE APPROVVIGIONA(
    id_ordine dom_id_ordine PRIMARY KEY,
    nome_fornitore varchar(50) NOT NULL,
    indirizzo_fornitore varchar(100) NOT NULL,
    nome_dipartimento varchar(50) NOT NULL,
    data date NOT NULL,
    FOREIGN KEY (nome_fornitore, indirizzo_fornitore) REFERENCES
    FORNITORE(nome_fornitore, indirizzo)
    ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (nome_dipartimento) REFERENCES DIPARTIMENTO(nome)
    ON UPDATE CASCADE ON DELETE CASCADE
);
```

```
CREATE TABLE LAVORA_A(
   matricola_dipendente dom_matricola NOT NULL,
   codice_progetto int NOT NULL,
   competenza varchar(50) NOT NULL,
   PRIMARY KEY (matricola_dipendente, codice_progetto, competenza),
   FOREIGN KEY (competenza, matricola_dipendente) REFERENCES
   HA_COMPETENZE(tipo_competenza, dipendente)ON UPDATE CASCADE ON DELETE
   CASCADE,
   FOREIGN KEY (codice_progetto) REFERENCES
   PROGETTO(codice_progetto_aziendale)ON UPDATE CASCADE ON DELETE CASCADE
);
```

5.3 Definizione di vincoli, trigger e funzioni

5.3.1 Vincoli

Sono stati definiti dei constraint che permettono di assicurare i vincoli, nello specifico:

1. il budget del progetto deve essere un numero compreso tra 1000 e 200.000,
2. la durata del progetto deve essere maggiore di 0 mesi,
3. l'e-mail e il telefono del dipartimento sono stati definiti univoci con la parola chiave UNIQUE.

Questi vincoli sono stati implementati direttamente nel codice SQL.

5.3.2 Trigger

All'interno della base di dati sono presenti altri tipi di vincoli che implicano la creazione di appositi trigger per essere rispettati:

1. il numero di matricola non può coincidere con il coniuge, dal momento che non è consentito sposarsi con se stessi.

```
CREATE OR REPLACE FUNCTION valida_matrimonio()
returns trigger AS
$$
 BEGIN
   PERFORM *
   FROM DIPENDENTE
   WHERE CAST(new.matricola AS int) = new.coniuge;
   IF FOUND
     then
       raise exception 'Matrimonio non consentito';
       return null;
     else
       return new;
     end if;
 END;
$$ language plpgsql;
```

```
CREATE trigger matrimonio
before insert or update on DIPENDENTE
for each row
execute procedure valida_matrimonio();
```

2. La data della laurea deve essere antecedente a quella del dottorato.

```
CREATE OR REPLACE FUNCTION valida_date_studi()
returns trigger AS
$$
 BEGIN
   PERFORM *
   FROM DIPENDENTE
   WHERE data_laurea < new.data_dottorato;

   IF FOUND
     then
       raise exception 'Data non conforme';
       return null;
     else
       return new;
     end if;
 END;
$$ language plpgsql;

CREATE trigger date_studi
before insert or update on DIPENDENTE
for each row
execute procedure valida_date_studi();
```

3. Abbiamo ipotizzato che l'azienda assuma solo persone maggiorenni, quindi la data di assunzione deve avere almeno 18 anni di differenza con la data di nascita.

```
CREATE OR REPLACE FUNCTION valida_maggiorenni()
returns trigger AS
$$
 BEGIN
   PERFORM *
   FROM DIPENDENTE
   WHERE DATE_PART('year', new.data_assunzione::date) -
   DATE_PART('year', new.data_di_nascita::date) < 18;

   IF FOUND
     then
       raise exception 'Data assunzione non conforme';
       return null;
     else
       return new;
     end if;
 END;
$$ language plpgsql;
```

```
CREATE trigger assunto_maggiorenne
before insert or update on DIPENDENTE
for each row
execute procedure valida_maggiorenni();
```

4. Se uno tra l'attributo *Coniuge* o *Data Matrimonio* è nullo, viene negato l'inserimento.

```
CREATE OR REPLACE FUNCTION valida_data_matrimonio_coniuge()
returns trigger AS
$$
 BEGIN
   PERFORM *
   FROM DIPENDENTE
   WHERE (new.coniuge is not null and new.data_matrimonio is null)
         or(new.coniuge is null and new.data_matrimonio is not
         null);
   IF FOUND
     then
       raise exception 'Attributo (coniuge o data) mancante nella
       relazione di matrimonio';
       return null;
     else
       return new;
     end if;
 END;
$$ language plpgsql;

CREATE trigger data_matrimonio_coniuge
before insert or update on DIPENDENTE
for each row
execute procedure valida_data_matrimonio_coniuge();
```

5.3.3 Funzioni

Dato che sono presenti attributi derivati abbiamo ritenuto opportuno implementarli utilizzando le user defined function.

1. La funzione calcola la data più recente per ricavare la data dell'ultimo acquisto.

```
CREATE OR REPLACE FUNCTION data_ultimo_acquisto(dip varchar)
RETURNS date
LANGUAGE plpgsql AS $$
 DECLARE
 data_max date;
 BEGIN
   SELECT max(data) INTO data_max
   FROM APPROVVIGIONA
   WHERE nome_dipartimento = dip;
   RETURN data_max;
 END
$$;
```

2. La funzione ricava il nome dell'ultimo fornitore in base alla data più recente dell'ultimo acquisto.

```
CREATE OR REPLACE FUNCTION nome_fornitore_ultimo_acquisto(data_max
date, dip varchar)
RETURNS varchar
LANGUAGE plpgsql AS $$
 DECLARE
 nome_ultimo_fornitore varchar;
 BEGIN
   SELECT nome_fornitore INTO nome_ultimo_fornitore
   FROM APPROVVIGIONA
   WHERE data = data_max and nome_dipartimento = dip;
   RETURN nome_ultimo_fornitore;
 END
$$;
```

3. La funzione ricava l'indirizzo dell'ultimo fornitore in base alla data più recente dell'ultimo acquisto.

```
CREATE OR REPLACE FUNCTION
indirizzo_fornitore_ultimo_acquisto(data_max date, dip varchar)
RETURNS varchar
LANGUAGE plpgsql AS $$
 DECLARE
 indirizzo_ultimo_fornitore varchar;
 BEGIN
   SELECT indirizzo_fornitore INTO indirizzo_ultimo_fornitore
   FROM APPROVVIGIONA
   WHERE data = data_max and nome_dipartimento = dip;
   RETURN indirizzo_ultimo_fornitore;
 END
$$;
```

5.4 Definizione di query significative

Di seguito vengono definite alcune query significative per la base di dati in analisi.

5.4.1 Query 1

I dipendenti assunti negli ultimi 12 mesi.

```
SELECT nome, cognome, data_assunzione
FROM DIPENDENTE
WHERE data_assunzione BETWEEN '2020-03-01' AND '2021-03-01';
```

5.4.2 Query 2

Il dipartimento che ha fatto più ordini negli ultimi 12 mesi.

```
CREATE VIEW max_a
AS
SELECT COUNT(id_ordine) AS nordine
FROM APPROVVIGIONA
GROUP BY nome_dipartimento;
```

```
SELECT A.nome_dipartimento, COUNT(A.id_ordine) AS nordine
FROM APPROVVIGIONA AS A
GROUP BY A.nome_dipartimento
HAVING COUNT(A.id_ordine) = (SELECT MAX(MA.nordine) FROM MAX_A AS MA);
```

5.4.3 Query 3

Il tipo di competenze che vengono applicate più di 5 volte.

```
SELECT LA.competenza, COUNT(LA.competenza) AS n_comp
FROM LAVORA_A AS LA
GROUP BY LA.competenza
HAVING COUNT(LA.competenza) > 5
ORDER BY COUNT(LA.competenza) desc;
```

5.4.4 Query 4

I dipendenti che hanno lavorato ad almeno tre progetti.

```
SELECT DISTINCT LA1.matricola_dipendente
FROM LAVORA_A AS LA1, LAVORA_A AS LA2, LAVORA_A AS LA3
WHERE LA1.matricola_dipendente=LA2.matricola_dipendente AND
      LA2.matricola_dipendente=LA3.matricola_dipendente AND
      LA1.codice_progetto<>LA2.codice_progetto AND
      LA1.codice_progetto<>LA3.codice_progetto AND
      LA3.codice_progetto<>LA2.codice_progetto;
```

6. PROGETTAZIONE FISICA

6.1 Studio degli indici

La creazione di strutture ausiliarie, quali gli indici, permette di migliorare le performance di un database. In particolare, gli indici vengono usati per migliorare i livelli di efficienza nella ricerca di record in una determinata tabella. Di base le chiavi primarie e gli attributi unici hanno già un indice, settato in automatico dal DBMS, quindi solitamente l'introduzione di indici viene effettuata su colonne e attributi non chiave.
La decisione d'inserimento di un indice non è banale, perchè non sempre un indice risulta utile nel miglioramento delle performance, in quanto se un certo campo di una tabella viene aggiornato spesso è fortemente sconsigliato l'utilizzo di un indice su di esso, perchè ad ogni modifica si dovrebbe modificare anche l'indice di conseguenza.
Un buon indice quindi deve avere delle caratteristiche che lo rendono tale, non deve essere troppo lungo e preferibilmente deve essere un numero, questo perché l'utilizzo di stringhe nelle operazioni di confronto implica l'utilizzo di un test di uguaglianza carattere per carattere che è piu lento, inoltre è consigliabile che abbia pochi valori duplicati altrimenti il lavoro dell'indice risulta altamente inefficace.

Nel nostro progetto dopo un'analisi generale non abbiamo ritenuto opportuno l'inserimento di un indice perchè non crediamo che le tabelle raggiungano un carico elevato, fatto esclusione per la tabella DIPENDENTE che anche se raggiungerà grandi dimensioni, sarà comunque possibile attraverso la chiave Matricola accedere velocemente al dipendente interessato. Un'ipotesi ragionevole poteva essere di inserire un indice sull'attributo *Nome* e *Cognome* ma nella ricerca di un dipendente solitamente si usa la matricola. Quindi crediamo che l'inserimento di un indice in questo caso sia irrilevante.

7. ANALISI DATI

7.1 Popolamento della base di dati

Per popolare la base di dati siamo partiti dalla tabella DIPENDENTE e successivamente abbiamo ricostruito le restanti tabelle. In particolare, abbiamo ipotizzato che l'azienda, e di conseguenza la base di dati, sia in funzione dal 2015 e che sia di piccole-medie dimensioni ma in crescita (50 dipendenti).

Per generare i dati delle tabelle abbiamo usato varie tecniche, ad esempio:

Per l'identità dei dipendenti (nome, cognome, data di nascita) è stato utilizzato il sito generatedata.com/#t1. Quest'ultimo offre anche una lista standard di nomi di dipartimento generalmente presenti in un'azienda, rendendo così più realistico il nostro database.

I nomi dei corsi di laurea e di dottorato sono stati importati principalmente dal sito dell'Università di Udine e in seguito da altri siti di università italiane.

DIPENDENTE

Per un corretto svolgimento del compito, ad alcuni dipendenti è stato assegnato un coniuge all'interno dell'azienda. Partendo dalla qualifica di ogni dipendente, abbiamo stabilito un eventuale corso di laurea ed un eventuale dottorato: si è presupposto che i dipendenti con qualifica *operaio* non siano in possesso di una laurea e di conseguenza nemmeno di un dottorato. Un'analisi minuziosa ha permesso che qualifica-laurea-dottorato-dipartimento di lavoro siano coerenti tra loro. Non vi si possono trovare infatti dipendenti laureti in economia che lavorano in IT. Tale coerenza è rispecchiata anche negli attributi data, come specificato già nei diversi vincoli e trigger; in particolare, per l'assegnazione dei coniugi, si è cercato di far combaciare le date di nascita dei due diversi dipendenti in modo tale da non ritrovarsi con un enorme disuguaglianza d'età.

PROGETTO

L'idea di base è stata quella di assegnare ad ogni progetto un numero di dipendenti che varia da 2 a 4. Sono stati identificati 15 progetti di durata diversa; il valore dell'attributo budget è stato conferito in base alla durata del progetto, con un valore massimo di 200'000€ per quello con durata maggiore.

DIPARTIMENTO

La lista dei dipartimenti è stata ricavata dal sito generatedata.com/#t1, come spiegato nelle righe precedenti.

COMPETENZE

Per creare una lista di competenze da assegnare ai diversi dipendenti, è stato necessario fare una ricerca online, che ha portato a considerare le soft e hard skills più richieste e comuni all'interno di un'azienda. L'assegnamento delle competenze è stato utile per la successiva attribuzione del dipartimento di appartenenza, citata in precedenza.

FORNITORE

I dati inerenti a tale tabella sono stati generati casualmente tramite il sito generatedata.com/#t1 che fornisce indirizzi e numeri di telefono di diverse nazioni.

Tutte le tabelle, una volta salvate in file ".csv", sono state caricate in PostgreSQL utilizzando il comando:

```
COPY nome_tabella FROM 'percorso_file\nome_file.csv' DELIMITER ';';.
```

7.2 Analisi dati in R

Infine, analizziamo il database in R.

7.2.1 Connessione al database

```
drv <- dbDriver('PostgreSQL')
con <- dbConnect(drv,
                 dbname='societa',
                 host='localhost',
                 port=5432,
                 user='postgres',
                 password='----------')
```

7.2.2 Query e rispettivi grafici

Prima richiesta: si vuole osservare quanti dipendenti sono presenti in ciascun dipartimento.
(presentiamo il codice solo della prima query e del primo grafico, le altre nei file allegati)

```
dipendenti_dipartimento<-dbGetQuery(con,
                                    "SELECT dipartimento, count(*)
                                    FROM dipendente
                                    GROUP BY dipartimento;")

grafico_dipendenti_dipartimento<-ggplot(dipendenti_dipartimento,
aes(dipartimento,count,fill = dipartimento))+
geom_col(show.legend = FALSE)+
coord_flip()+
labs(x = 'Dipartimento', y = 'Numero dipendenti', title = 'NUMERO DIPENDENTI
PER DIPARTIMENTO')+
theme_classic()+
scale_y_continuous(breaks = pretty_breaks())

grafico_dipendenti_dipartimento
```

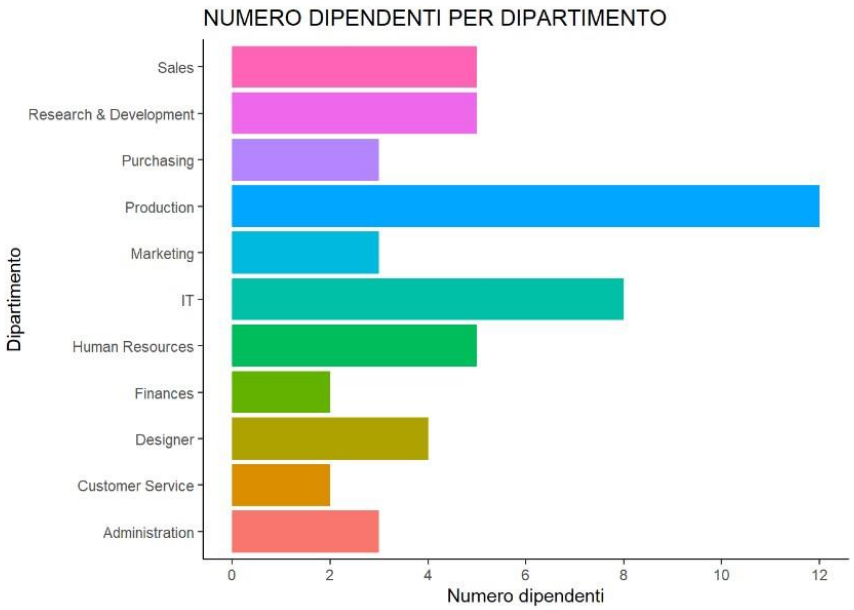

Grafico 1 - numero dipendenti per dipartimento

Il dipartimento *Production* risulta essere quello con più dipendenti, seguito da *IT*.

Seconda richiesta: si vuole esaminare l'andamento delle assunzioni all'interno dell'azienda per ogni anno.

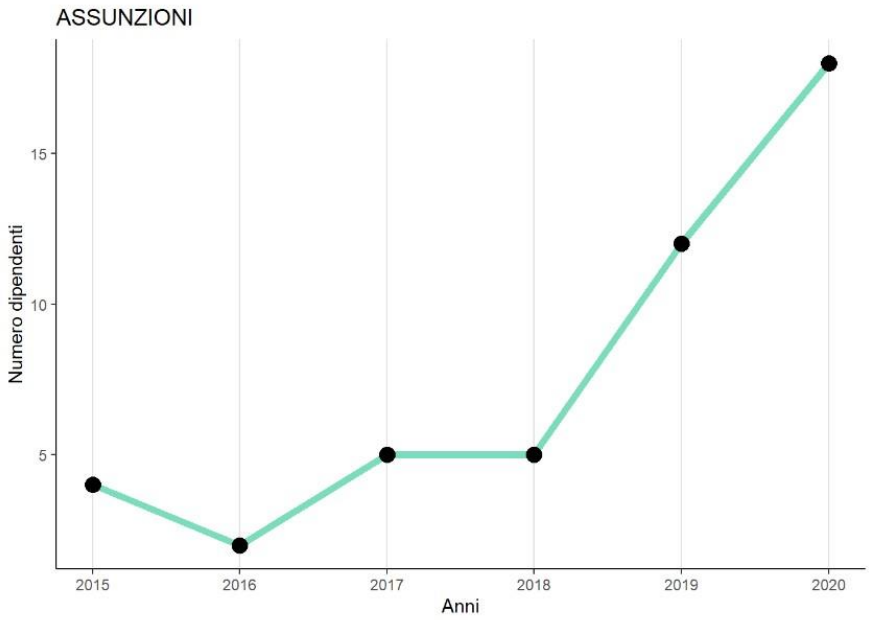

Grafico 2 - andamento assunzioni

Il grafico conferma che la società è in netta crescita, come annunciato in precedenza.

Terza richiesta: si vuole osservare come cambia il budget di un progetto in base alla sua durata, in particolare si vuole capire se c'è un qualche tipo di relazione tra queste due variabili oppure se la correlazione è casuale.

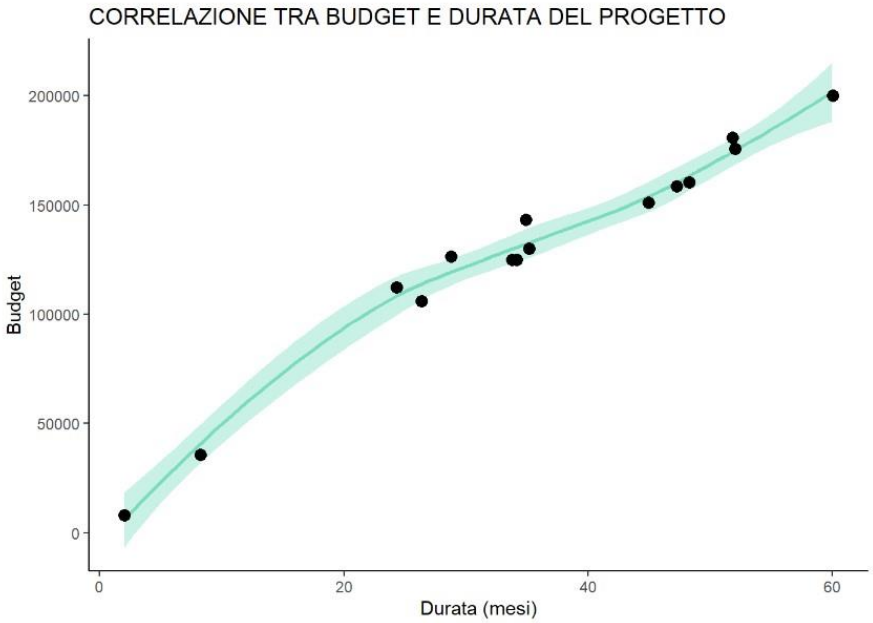

Grafico 3 - correlazione budget/durata

Dallo studio del grafico si può dedurre che tra il budget di un progetto e la sua durata c'è una correlazione che tende alla linearità, ovvero all'aumentare della durata aumenta anche il budget del progetto.

Quarta richiesta: si vuole illustrare il numero di competenze utilizzate per ciascun progetto.

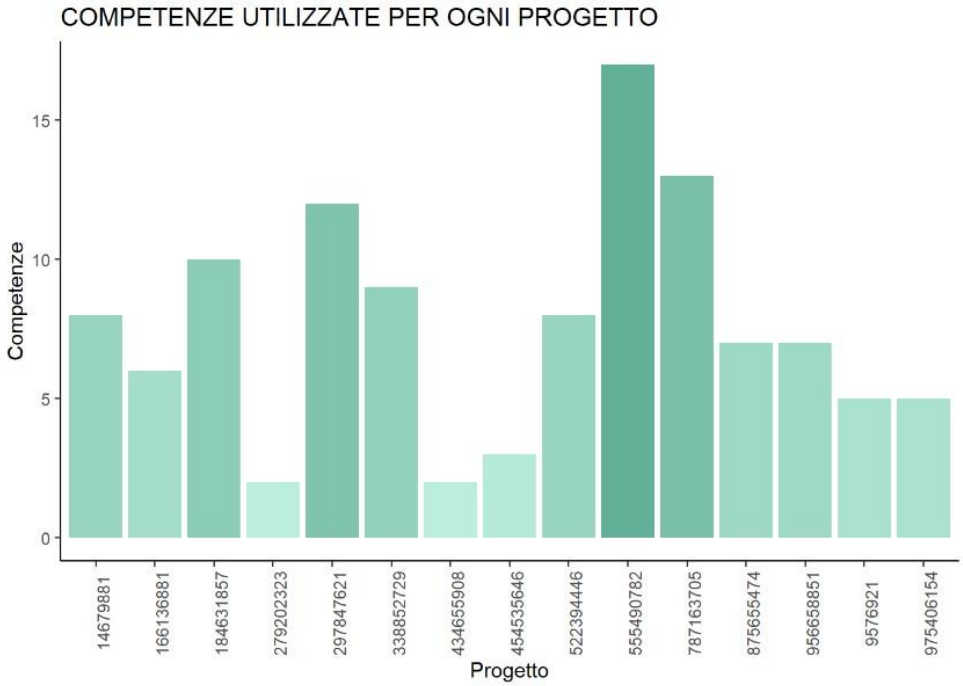

Grafico 4 - competenze utilizzate per progetto

Il progetto “555490782” è quello in cui vengono utilizzate il maggior numero di competenze ovvero 17. Da una visione globale si può notare che vengono utilizzate tipicamente molte competenze, difatti solo in tre progetti vengono utilizzate meno di quattro competenze.