

Relazione per il corso Basi di Dati

A.A. 2017/2018

# PROGETTO DI UNA BASE DATI PER LA GESTIONE DI UNO STUDIO COMMERCIALE

---

GRUPPO: 1551

STUDENTI:

Corina Jacopo ([jacopo.corina@studio.unibo.it](mailto:jacopo.corina@studio.unibo.it)) - 792482

Galdenzi Federico ([federico.galdenzi@studio.unibo.it](mailto:federico.galdenzi@studio.unibo.it)) - 801886

Rossolini Andrea ([andrea.rossolini2@studio.unibo.it](mailto:andrea.rossolini2@studio.unibo.it)) - 794193

Università degli studi di Bologna

Campus di Cesena

Facoltà di Ingegneria e Scienze Informatiche

## *Sommario*

1 – Introduzione .....	2
2 - Analisi dei requisiti .....	2
2.1 – estrazione dei concetti principali .....	2
2.2–note sulle caratteristiche del dominio .....	3
2.2.1 - Analisi della figura del CLIENTE e del RESPONSABILE .....	3
2.2.2 - Analisi della figura del DIPENDENTE e della FASE.....	4
3 – Progettazione concettuale .....	5
3.1 - tabella generale .....	5
3.2 – Composizione dello schema scheletro .....	6
3.2.1–Clienti.....	6
3.2.2 – Prestazioni e fasi.....	7
3.3 – SCHEMA FINALE .....	8
4 – Progettazione logica.....	9
4.1 – Stima del volume dei dati .....	9
4.2 – Descrizione delle operazioni principali .....	10
4.3 – Schemi di navigazione e tabelle degli accessi .....	13
4.4 – Analisi delle ridondanze .....	24
4.5 – Raffinamento dello schema .....	27
4.6 – Traduzione delle entità e associazioni in relazioni .....	31
4.7 – Schema logico finale .....	32
4.8 – Stesura delle principali operazioni in linguaggio SQL .....	32
4.9 – Inserimento di vincoli generici .....	37
4.10 – Inserimento dei trigger.....	39
4.10.1 – trigger per l'entità FASE .....	39
4.10.2 – trigger per l'entità RESPONSABILE.....	40
4.10.3 – trigger per l'entità PRATICA.....	41
4.10.4 – trigger per l'entità PRESTAZIONE .....	42
5 – Progettazione dell'applicazione.....	43
5.1 - Descrizione dell'architettura dell'applicazione realizzata .....	43
5.1.1 - Form di login:.....	43
5.1.2 - Schermata principale:.....	44
5.1.3 - Schermata di inserimento costi struttura: .....	45
5.1.4 - Schermata Form per dipendente:.....	46
5.1.5 - Schermata Amministratore Inserimenti: .....	46
5.1.6 - Schermata di Interrogazioni per l'amministratore: .....	47

## 1 – Introduzione

*“Una base dati per uno studio professionale che permetta una rendicontazione di tutte le attività svolte dallo studio nei confronti dei clienti con in evidenza il tipo di attività, i tempi di realizzazione della prestazione in tutte le sue fasi, quindi dall'avvio al termine della prestazione e infine la sua valorizzazione sia in base al tempo impiegato dai professionisti, collaboratori e dipendenti che in base alle tariffe applicabili al tipo di prestazione professionale e o di servizi.*

*Obiettivi:*

*A) rendiconto al cliente delle singole attività con i tempi di realizzazione e il singolo compenso dovuto;*

*B) rendiconto al cliente di tutte le attività effettuate in un determinato periodo (anno solare) con i tempi a lui dedicati sia complessivi che suddivisi tra professionisti, collaboratori e dipendenti e dei compensi dovuti.*

*C) rendiconto per professionista, collaboratore e dipendente delle attività svolte, con valorizzazione dei compensi percepiti dallo studio e valorizzazione dei costi sostenuti dallo studio, sia per la remunerazione dei costi del lavoro che dei costi di struttura;*

*D) rendiconto per cliente delle attività svolte con valorizzazione dei compensi percepiti dallo studio e dei costi sostenuti dallo studio con indicati o né delle marginalità in valori monetari e percentuali.”*

## 2 - Analisi dei requisiti

### 2.1 – estrazione dei concetti principali

*Si vuole realizzare un database per uno studio commerciale che permetta la memorizzazione dei clienti e dei dipendenti, catalogando anche la rendicontazione delle attività svolte da ogni dipendente.*

*Lo studio ha dei clienti che, tramite la richiesta di una pratica, domandano allo studio una serie di prestazioni. Dei clienti, che possono essere aziende o persone fisiche, si memorizzano il codice fiscale, un nominativo ed il loro indirizzo, eventualmente anche una partita IVA se si tratta di un'azienda; la richiesta verrà effettuata in una determinata sede, della quale si vuole memorizzare il nome, la città e la via.*

*I dipendenti, dei quali si memorizza il rispettivo ruolo, il numero di matricola, i dati anagrafici ed il loro costo orario, si dividono in impiegati, interni allo studio ed in collaboratori, che possono essere anche esterni allo studio; i primi avranno*

*un loro stipendio, derivato dal costo orario; i dipendenti potranno essere allocati in una o più sedi. Le prestazioni richieste dai clienti possono essere svolte in più fasi, dove, almeno una, è quella iniziale; di queste fasi si vuole memorizzare il dipendente che la esegue, la prestazione a cui fa riferimento, il tempo impiegato per eseguirla, tramite un orario di inizio e di fine, ed una breve descrizione.*

*Gli impiegati possono prendere sotto la propria responsabilità, ovvero come se fossero delle persone di riferimento, alcuni clienti, ma solo se il cliente che richiede la responsabilità lo fa nella sede dove il dipendente è allocato.*

*Le prestazioni commerciali, delle quali si vuole catalogare un codice, il compenso preventivato e si vorrà sapere se la suddetta prestazione è stata terminata e pagata (alcune prestazioni possono anche essere pagate prima di essere terminate), fanno riferimento ad una categoria di attività commerciali, che hanno un codice, un nome, un costo fisso ed una breve descrizione, alcune delle quali esclusive per aziende o persone fisiche, fanno parte, a loro volta, di macro categorie che le raggruppano.*

*Lo studio inoltre deve sopperire a dei costi di mantenimento che variano da costi mensili (come bollette, acqua ecc...), a costi materiali, dei quali si vuole memorizzare la quantità di tali materiali e costi imprevisti.*

## **2.2-note sulle caratteristiche del dominio**

Nonostante la descrizione sia piuttosto esaustiva procediamo con l'analisi più dettagliata delle figure principali all'interno del dominio.

### **2.2.1 - Analisi della figura del CLIENTE e del RESPONSABILE**

I clienti, divisi in persone fisiche e aziende dotate di partita IVA, si presentano ad una sede dello studio per richiedere una serie di prestazioni commerciali. Il cliente non ha alcun interesse a richiedere informazioni riguardo alle sottocategorie delle prestazioni richieste, ai costi di struttura legati alla sede in cui richiede il servizio ed al numero di fasi in cui viene terminata la prestazione. Il dovere del cliente è quella di richiedere una o più prestazioni e di pagare lo studio in base a quanto gli viene richiesto per ogni prestazione.

Al cliente può essere legata, in caso di necessità, una figura di riferimento, ovvero il responsabile, il quale è un impiegato (quindi un dipendente) interno allo studio. Per divenire responsabile di un cliente, il dipendente interessato dovrà obbligatoriamente essere allocato nella sede in cui il cliente richiede la pratica. Il responsabile può essere cambiato nel corso del tempo, anche in caso la pratica richiesta dal cliente non fosse terminata, per ragioni interne allo studio o anche per ragioni esterne; per questo motivo, per la figura del

responsabile, il dato che rappresenterà (tramite una data) la fine della responsabilità sarà un attributo opzionale, pertanto, in caso venga conclusa una responsabilità, sempre se ce ne fosse ancora la necessità, la data che indicherà l'inizio della nuova responsabilità per un impiegato dovrà essere coincidente con la data di fine della responsabilità dell'impiegato che lo ha preceduto.

### ***2.2.2 - Analisi della figura del DIPENDENTE e della FASE***

Come già riportato nella descrizione del dominio, la figura del dipendente può essere divisa nella figura dell'impiegato, ovvero un dipendente interno allo studio, e nella figura del collaboratore, un lavoratore che può essere esterno allo studio.

Le due figure professionali sono distinte dal fatto che il dipendente, essendo interno allo studio, avrà uno stipendio fisso, nonostante questo sia derivabile dal costo orario, al contrario il collaboratore verrà retribuito in base alle ore dedicate al lavoro relazionato allo studio.

Infatti, i collaboratori possono svolgere, esattamente come gli impiegati, le fasi per le prestazioni richieste dai clienti, magari in collaborazione (appuntamento) con i membri interni dello studio.

Come spiegato nella descrizione del dominio le prestazioni vengono svolte in varie fasi, necessariamente è presente una fase iniziale, che può essere anche quella conclusiva, in caso si tratti di una prestazione non troppo complicata, ma una prestazione può estendersi anche in diverse fasi (in alcuni casi anche piuttosto numerose). Di queste fasi si memorizzano gli orari di inizio e di fine, con una breve descrizione del lavoro svolto, in questo modo sarà facile ricavare la durata generale dedicata allo svolgimento di una fase e quindi di un'intera prestazione.

### 3 – Progettazione concettuale

Si procede ora con lo sviluppo dello schema concettuale. Si procederà dunque per diverse fasi più o meno indipendenti tra di loro.

#### 3.1 - tabella generale

Di seguito è mostrata la tabella che indica il nome delle entità e delle relazioni, con, a fianco, una descrizione riassuntiva

CONCETTO	TIPO	DESCRIZIONE
RUOLO	E	Ruolo ricoperto da un dipendente
DIPENDENTE	E	Dipendente che sta lavorando all'azienda
COLLABORATORE	E	Eventuale collaboratore, interno od esterno all'ufficio, che può che lavora insieme al dipendente ad una prestazione
RESPONSABILE	E	Impiegato di riferimento per un determinato cliente
IMPIEGATO	E	Impiegato, appartenente allo studio, che svolge la prestazione per un cliente
SEDE	E	Eventuali sedi dello studio
COSTO_STRUTTURA	E	Costi che lo studio deve sostenere
PERIODICO	E	volume dei costi valutati ogni mese per un anno (utenze, affitto, ecc...)
IMPREVISTO	E	Costi non preventivati dallo studio (come un guasto tecnico ad un pc od eventuali costi di ristrutturazione)
MATERIALE	E	Materiali utilizzati in ufficio (calcolatrici, penne, fogli, toner ecc..)
CLIENTE	E	Dati del cliente che richiede una pratica allo studio
PERSONA FISICA	E	Rappresenta le informazioni di un cliente come persona fisica
AZIENDA	E	Rappresenta le informazioni di un cliente come azienda
PRESTAZIONE	E	Riporta le prestazioni che compongono il preventivo richiesto dal cliente
PRATICA	E	Pratica richiesta dal cliente che riporta le attività richieste
FASE	E	Numero di fasi che compongono una prestazione
SOTTO_CATEGORIA	E	Specializzazione della categoria

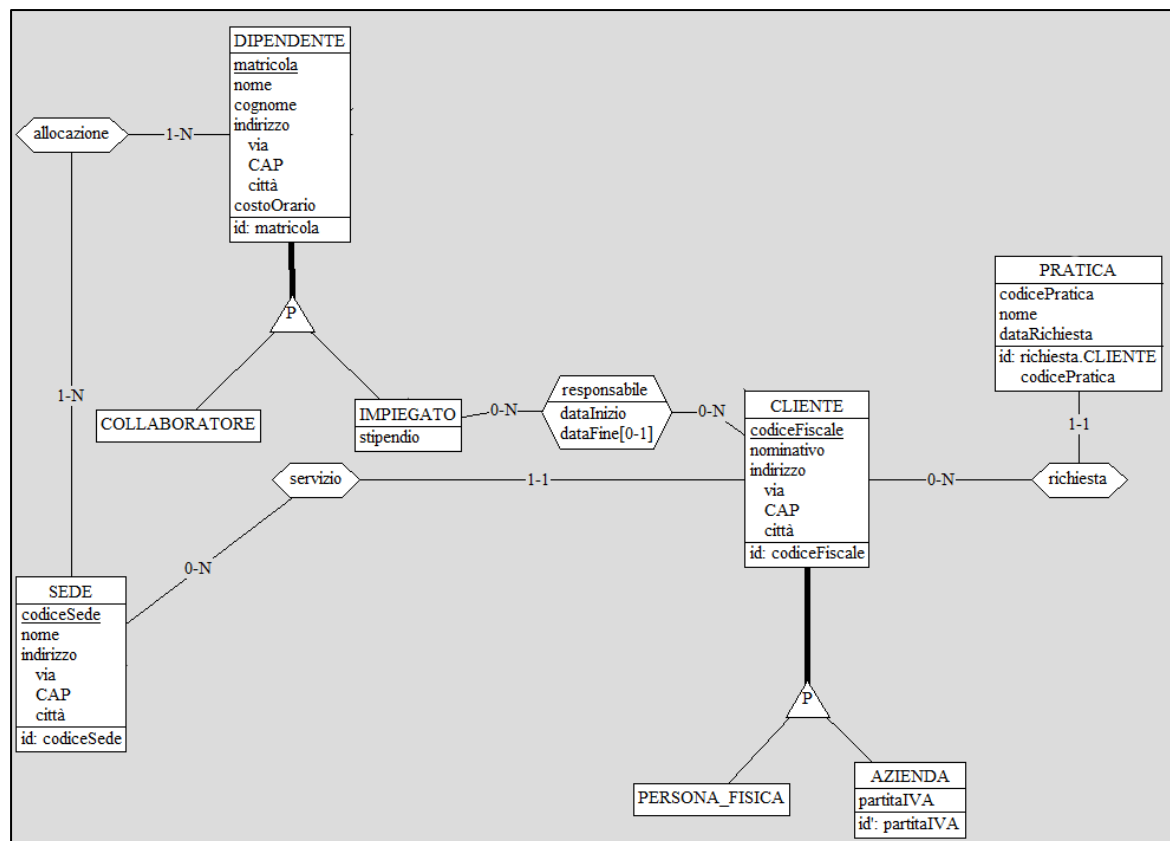
SOTTOCATEGORIA_PF	E	Attività che possono essere richieste esclusivamente da persone fisiche
SOTTOCATEGORIA_AZ	E	Attività che possono essere richieste esclusivamente da aziende
CATEGORIA	E	Categoria ove ricade l'attività richiesta dal cliente
ricopre	A	Lega il dipendente al ruolo che ricopre
allocazione	A	Lega il dipendente alla sede in cui risiede
salda	A	Lega la sede ai suoi relativi costi di struttura
esecuzione	A	Lega il dipendente con una determinata fase di una prestazione
attribuzione	A	Lega un dipendente al compito di responsabile
assistenza	A	Lega un cliente al responsabile che lo serve
suddivisione	A	Lega le prestazioni alle fasi in cui è suddivisa
tipologia	A	Lega la prestazione alla sottocategoria a cui appartiene
composizione	A	Lega la pratica alle prestazioni richieste dal cliente
servizio	A	Lega il cliente alla sede in cui richiede il servizio
richiesta	A	Lega il cliente alla pratica che richiede
specializzazione	A	Lega la sottocategoria alla categoria principale a cui appartiene

### **3.2 – Composizione dello schema scheletro**

In base alle informazioni estrapolate dai paragrafi precedenti procediamo alla costruzione di un primo scheletro dello schema concettuale

#### **3.2.1–Clienti**

Quindi dopo aver esaminato il dominio, viene proposto il seguente schema:



Si può notare che la relazione “responsabile” è conveniente reificarla, dato che si trova in una relazione N-M, poiché un cliente può avere più responsabili e un impiegato può essere responsabile di più clienti, ed inoltre è di nostro interesse memorizzare la data di inizio e la data di fine della responsabilità di un impiegato.

La relazione con sede è necessaria, poiché un cliente non può avere come responsabile un cliente che non appartiene alla sede nella quale il cliente richiede il servizio.

Si può notare come l’entità pratica sia direttamente dipendente dal cliente che la richiede.

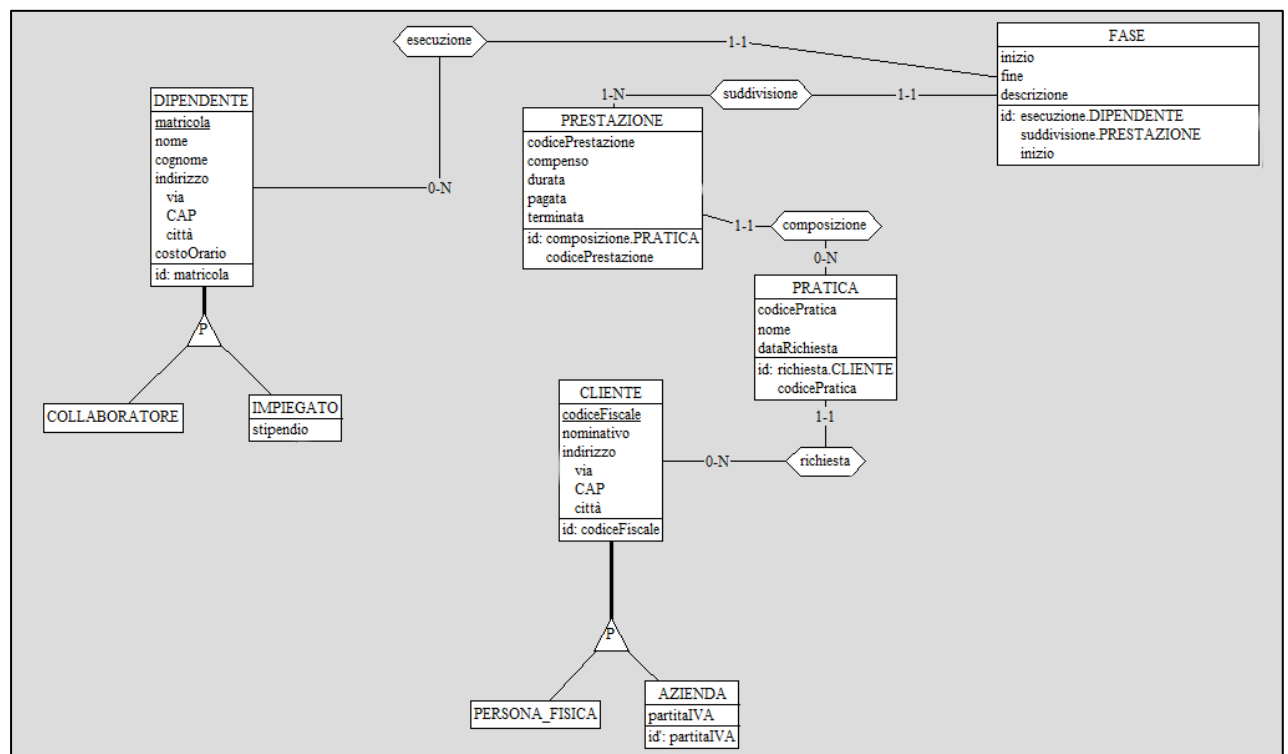
### 3.2.2 – Prestazioni e fasi

Presumendo che ci sia una dipendenza diretta tra la pratica richiesta dal cliente ed il cliente stesso e tra una fase di una determinata prestazione e la medesima, abbiamo deciso di far dipendere la prestazione, che compone la pratica e che quindi mette in relazione le fasi con la pratica succitata, dalla pratica alla quale è associato.

Inoltre le fasi di una prestazione sono identificate anche dal dipendente, o collaboratore, che la svolge.

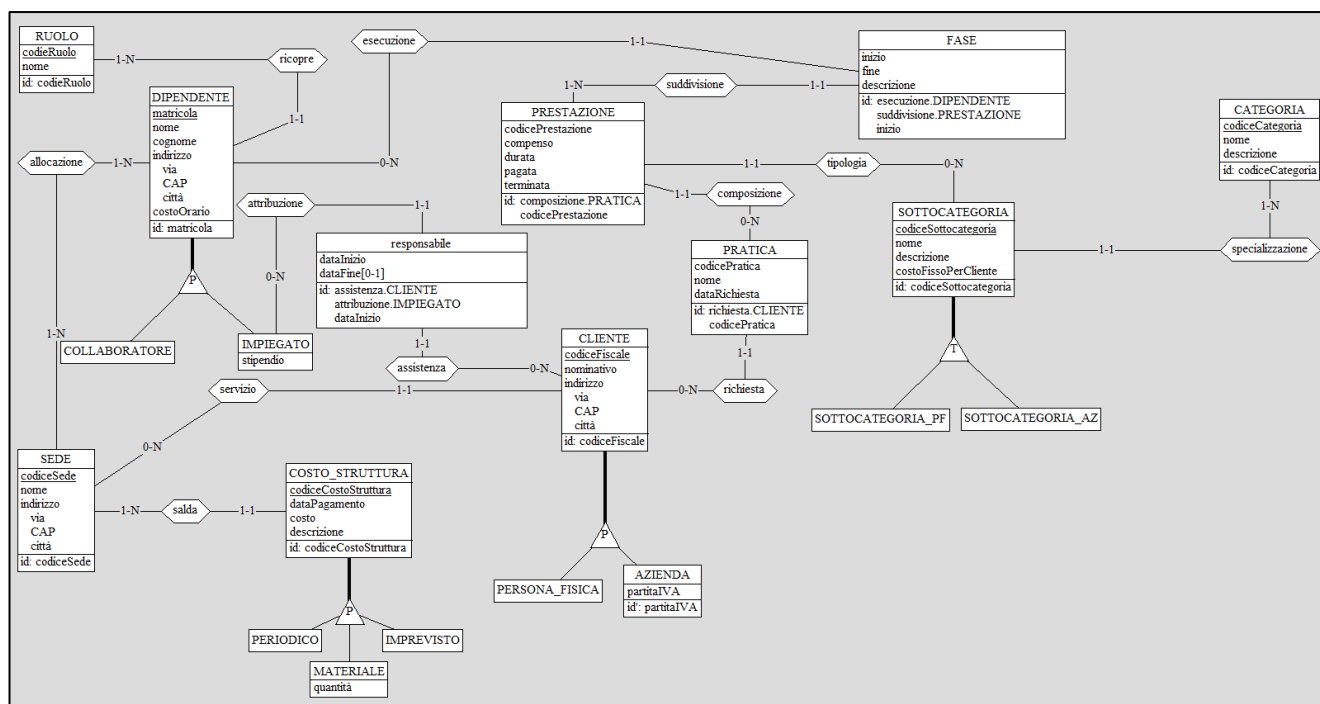
Lo schema risulterà dunque:





### 3.3 – SCHEMA FINALE

Di seguito è riportato lo schema finale, dove si palesano anche le altre entità che erano state menzionate in precedenza, ma che non sono state ulteriormente approfondite; come ad esempio i costi struttura o le varie categorie e sottocategorie (sufficientemente delineate nell'analisi del dominio).



## 4 – Progettazione logica

### 4.1 – Stima del volume dei dati

Di seguito è riportata una tabella che specifica il numero stimato di istanze per ogni entità (E) e per associazione (A) dello schema.

CONCETTO	COSTRUTTO	VOLUME
RUOLO	E	10
DIPENDENTE	E	20
COLLABORATORE	E	5
RESPONSABILE	E	100
IMPIEGATO	E	15
SEDE	E	2
COSTO_STRUTTURA	E	110
PERIODICO	E	60
IMPREVISTO	E	20
MATERIALE	E	30
CLIENTE	E	100
PERSONA FISICA	E	30
AZIENDA	E	70
PRESTAZIONE	E	7000
PRATICA	E	1500
FASE	E	21000
SOTTO_CATEGORIA	E	40
SOTTOCATEGORIA_PF	E	10
SOTTOCATEGORIA_AZ	E	30
CATEGORIA	E	15
ricopre	A	20
allocazione	A	20
salda	A	110
esecuzione	A	21000
attribuzione	A	100
assistenza	A	100
suddivisione	A	21000
tipologia	A	7000
composizione	A	7000
servizio	A	100
richiesta	A	1500
specializzazione	A	40

#### ***4.2 – Descrizione delle operazioni principali***

Nella tabella sottostante vengono mostrate le principali operazioni richieste, con a fianco la loro frequenza di esecuzione. Le varie operazioni verranno poi approfondite nel corso del paragrafo, con relativa tabella degli accessi e schema di navigazione. Non verranno approfondite tali operazioni che coinvolgono una sola tabella oppure che presentano uno schema di navigazione identico ad uno già presentato.

OPERAZIONI DI INSERIMENTO		
Codice	Descrizione	Frequenza
1	Inserimento di una fase	200/giorno (10 per ogni dipendente)
2	Inserimento dei costi di struttura	1/Mese
3	Inserimento di un dipendente *	1/Anno
4	Inserimento di un cliente*	3/Mese
5	Inserimento di una pratica	2/settimana
6	Inserimento di una prestazione	3/settimana
7	<i>Operazioni a frequenza indipendente</i>	
7.1	Inserimento di un ruolo *	
7.2	Inserimento di una sede *	
7.3	Inserimento di una categoria *	
7.4	Inserimento di una sottocategoria *	

(\*) operazioni riservate all'amministratore.

Codice	Descrizione	Frequenza
V1	Visualizzare tutte le fasi svolte per un determinato cliente e info su di esso, sulla prestazione e sulla pratica*	1/Mese
V2	Visualizzare compensi e dettagli di una certa pratica richiesta da un cliente	16/Mese
V3	Visualizzazione di tutte le fasi di una prestazione	15/Mese
V4	Visualizzare tutte le prestazioni terminate e/o pagate in base a un determinato cliente*	1/Giorno
V5	Visualizzare per ogni cliente il relativo storico dei responsabili*	1/Mese

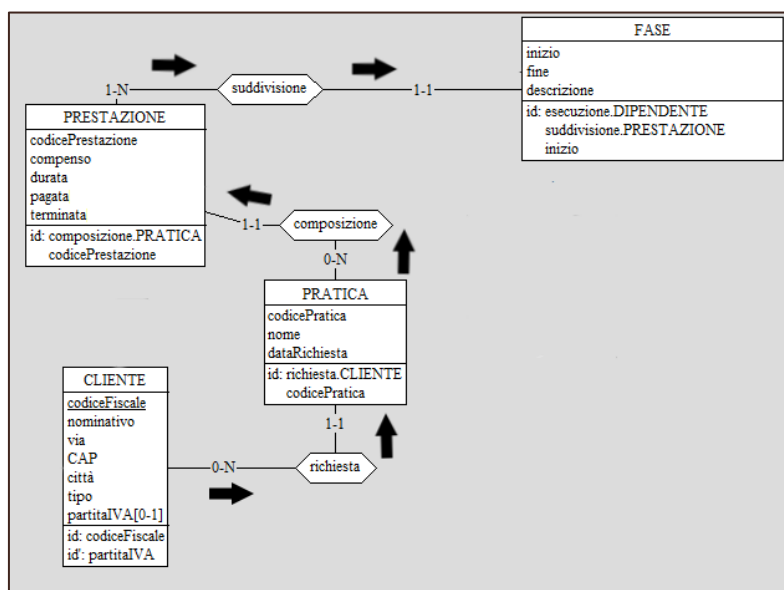
V6	Visualizzare per ogni sede i relativi costi struttura*	1/Mese
V7	Visualizzare il totale delle ore dedicate ad un cliente*	2/Mese
V8	Visualizzare dei 5 dipendenti che hanno fatto meno fasi*	1/Mese
V9	Visualizzare la sottocategoria (e la relativa categoria) più richiesta*	1/Mese
V10	Visualizzare le prestazioni la cui durata supera la durata media delle prestazioni*	1/Mese
V11	Visualizzare i dipendenti e le relative allocazioni*	1/Mese
V12	Visualizzare per una certa sede i dipendenti ed il relativo ruolo*	10/Mese
V13	Visualizzare l'elenco dei clienti di una certa tipologia*	1/Mese
V14	Visualizzare lo stipendio di un dipendente*	1/Mese
V15	Visualizzare per ogni sede le città dei relativi clienti*	1/Mese
V16	Visualizzare le prestazioni per un cliente eseguite da un determinato dipendente	1/Mese
V17	Visualizzare la durata complessiva di una prestazione	4/Mese
V18	Visualizzare il numero di* prestazioni pagate per cliente in ordine decrescente	4/Mese
V19	Visualizzare i dipendenti e le relative responsabilità nel tempo, visualizzando anche chi non le ha mai avute*	1/mese

### 4.3 – Schemi di navigazione e tabelle degli accessi

V1 - Visualizzare tutte le fasi svolte per un determinato cliente e info su di esso, sulla prestazione e sulla pratica

CONCETTO	COSTRUTTO	ACCESSI	TIPO
CLIENTE	E	1	L
richiesta	R	15	L
PRATICA	E	15	L
composizione	R	75	L
PRESTAZIONE	E	75	L
suddivisione	R	225	L
FASE	E	225	L

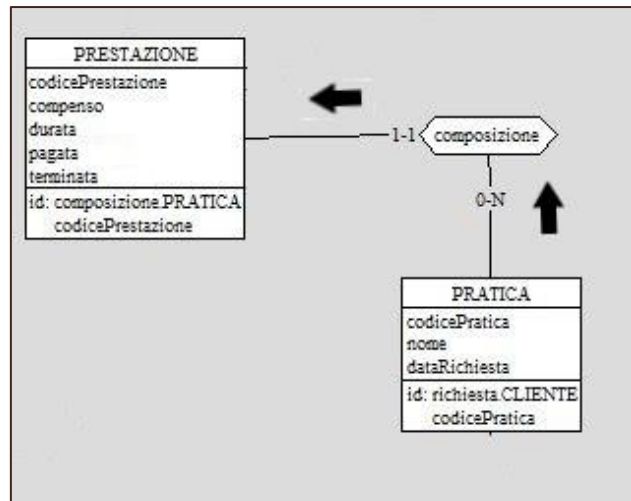
TOTALE:	631	* 1/mese	631/mese
---------	-----	----------	----------



V2 - Visualizzare compensi e dettagli di una certa pratica richiesta da un cliente

CONCETTO	COSTRUTTO	ACCESSI	TIPO
PRATICA	E	1	L
composizione	A	5	L
PRESTAZIONE	E	5	L

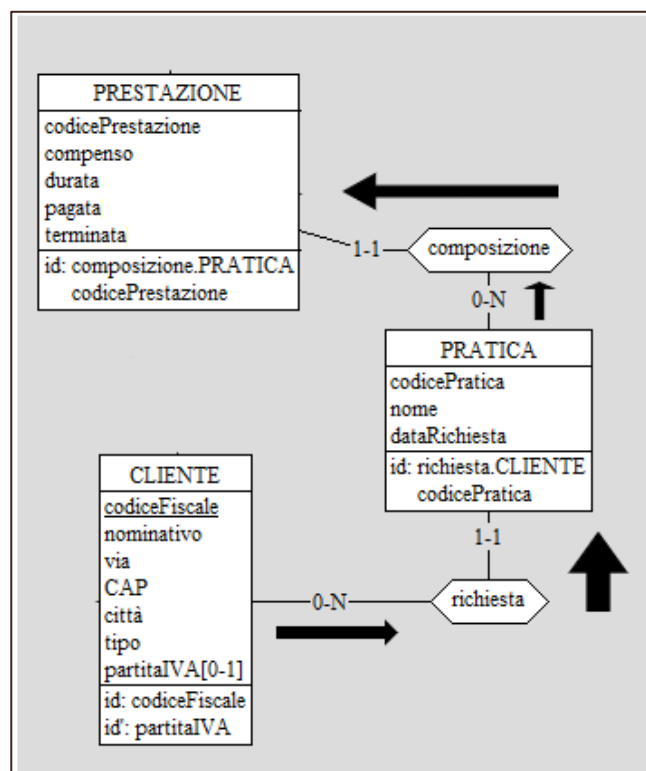
TOTALE:	11	*16/mese	176/mese
---------	----	----------	----------



V4 - Visualizzare tutte le prestazioni terminate e/o pagate in base a un determinato cliente

CONCETTO	COSTRUTTO	ACCESSO	TIPO
CLIENTE	E	1	L
richiesta	A	15	L
PRATICA	E	15	L
composizione	A	70	L
PRESTAZIONE	E	70	L

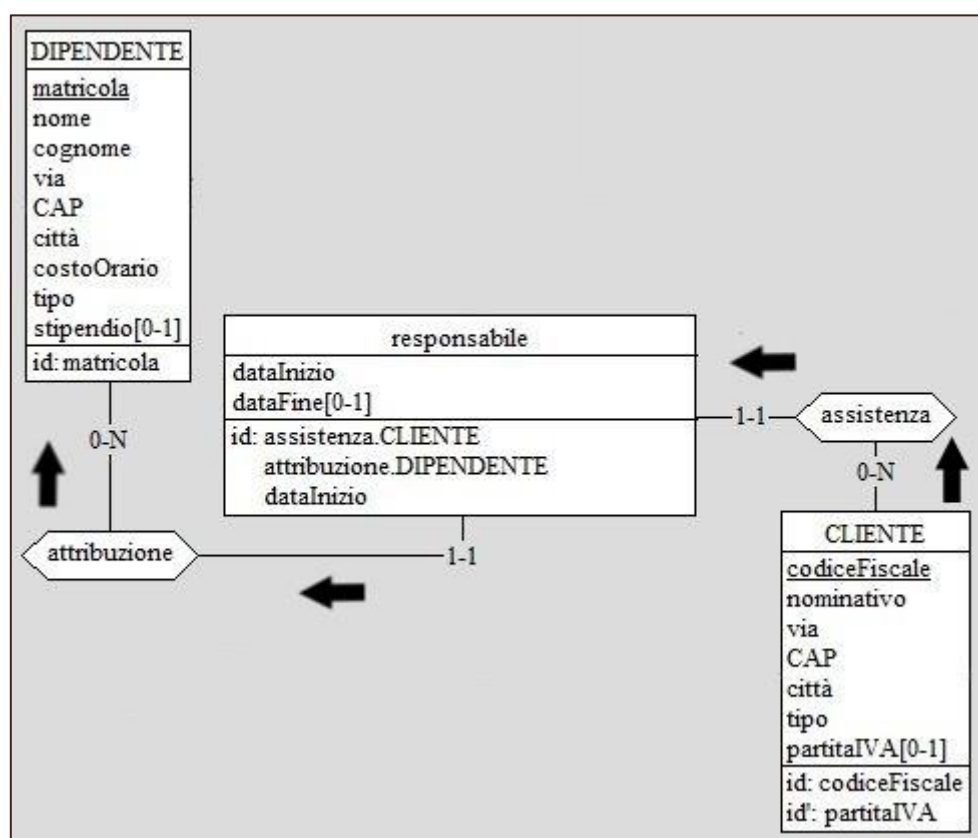
TOTALE:	171	*1/mese	171/mese
---------	-----	---------	----------



V5 - Visualizzare per ogni cliente il relativo storico dei responsabili

CONCETTO	COSTRUTTO	ACCESSI	TIPO
CLIENTE	E	100	L
assistenza	A	100	L
RESPONSABILE	E	100	L
attribuzione	A	100	L
DIPENDENTE	E	100	L

TOTALE:	500	*1/mese	500/mese
---------	-----	---------	----------

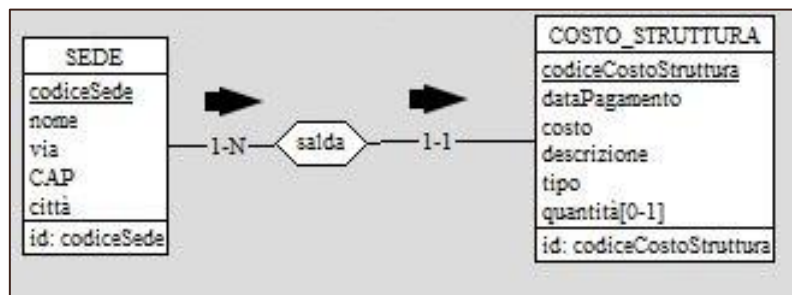




V6 - Visualizzare per ogni sede i relativi costi struttura

CONCETTO	COSTRUTTO	ACCESSI	TIPO
SEDE	E	2	L
salda	A	110	L
COSTO_STRUTTURA	E	110	L

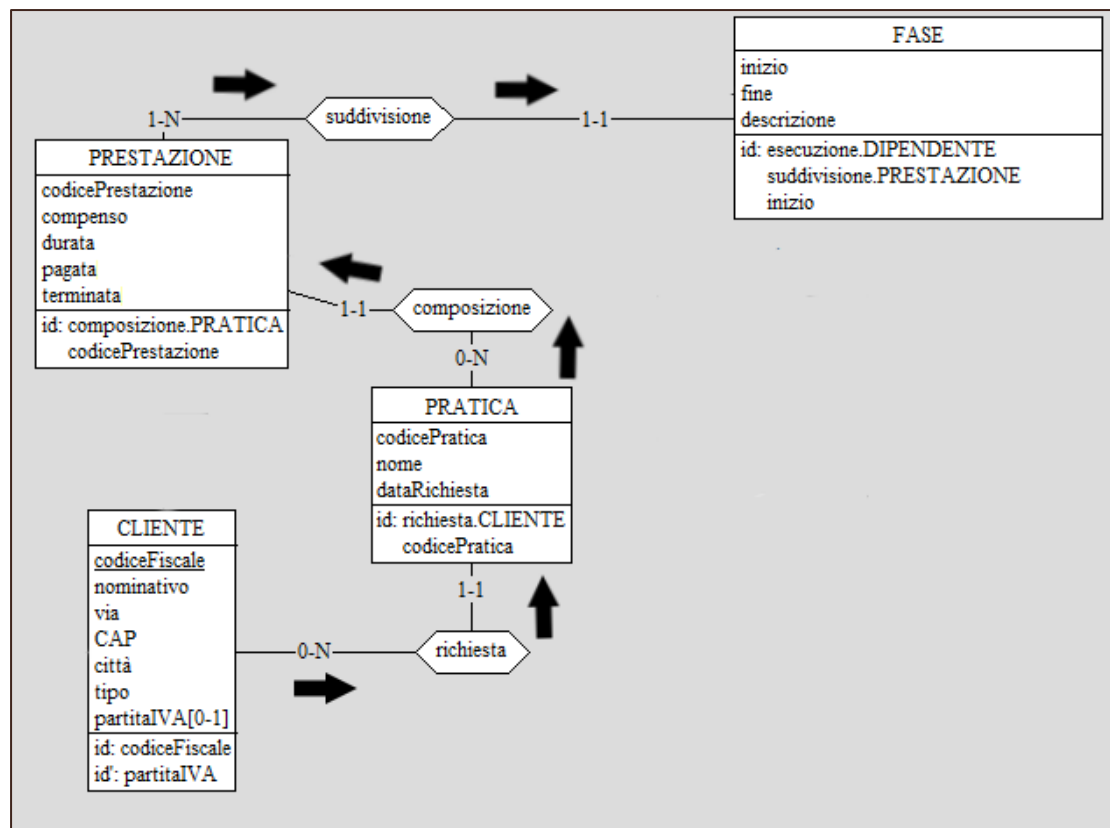
TOTALE:	222	1/mese	222/mese
---------	-----	--------	----------



V7 - Visualizzare il totale delle ore dedicate ad un cliente

CONCETTO	COSTRUTTO	ACCESSO	TIPO
CLIENTE	E	1	L
richiesta	A	$1500/100 = 15$	L
PRATICA	E	15	L
composizione	A	$15 \cdot (7000/1500) = 75$	L
PRESTAZIONE	E	75	L
suddivisione	A	$75 \cdot (21000/7000) = 225$	L
FASE	E	225	L

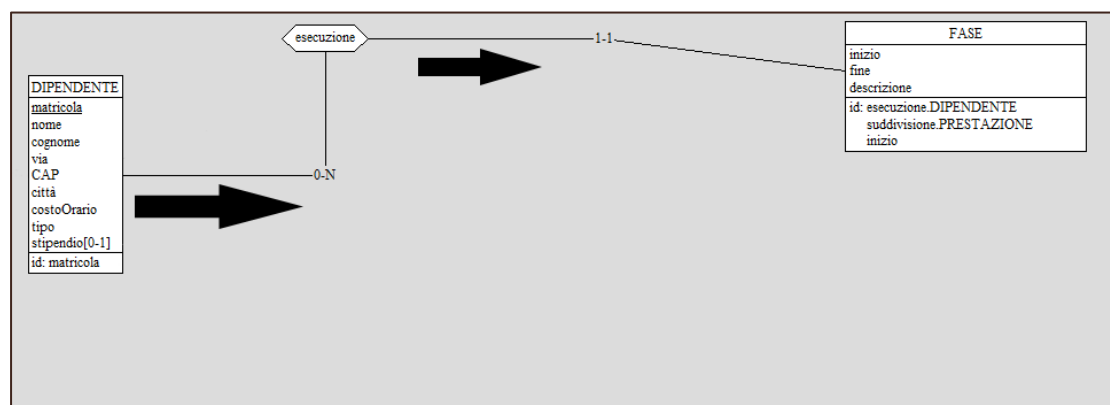
TOTALE:	631	4/mese	2524/mese
---------	-----	--------	-----------



V8 - Visualizzare dei 5 dipendenti che hanno fatto meno fasi

CONCETTO	COSTRUTTO	ACCESSO	TIPO
DIPENDENTE	E	20	L
esecuzione	A	21000	L
FASE	E	21000	L

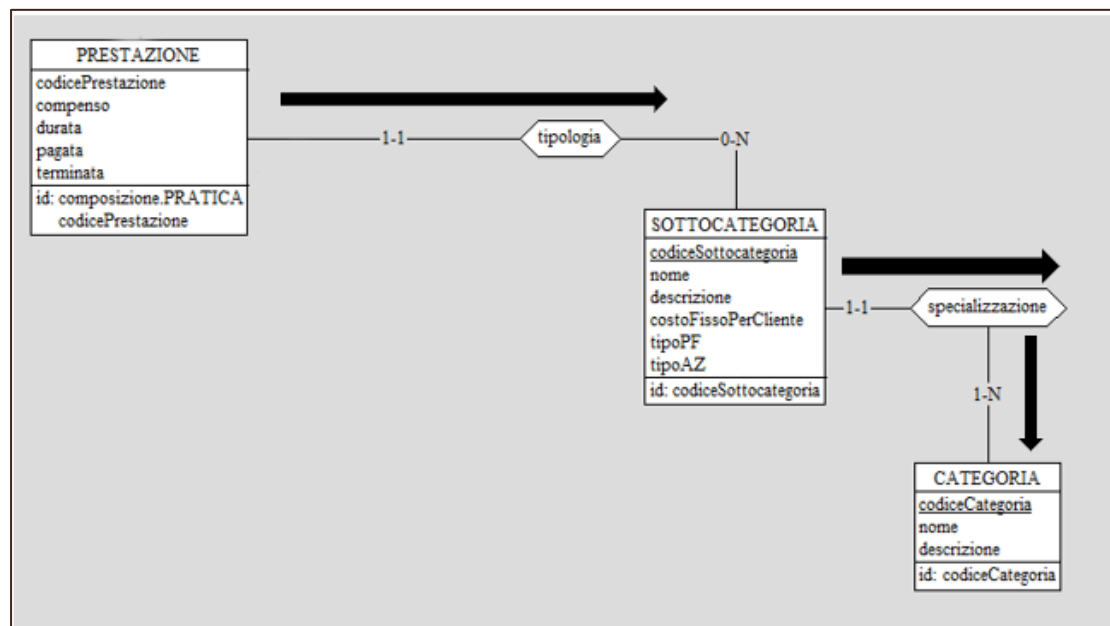
TOTALE	42020	*1/mese	42020/mese
--------	-------	---------	------------



V9 - Visualizzare la sottocategoria (e la relativa categoria) più richiesta

CONCETTO	COSTRUTTO	ACCESSO	TIPO
PRESTAZIONE	E	7000	L
tipologia	A	7000	L
SOTTOCATEGORIA	E	7000	L
specializzazione	A	7000	L
CATEGORIA	E	7000	L

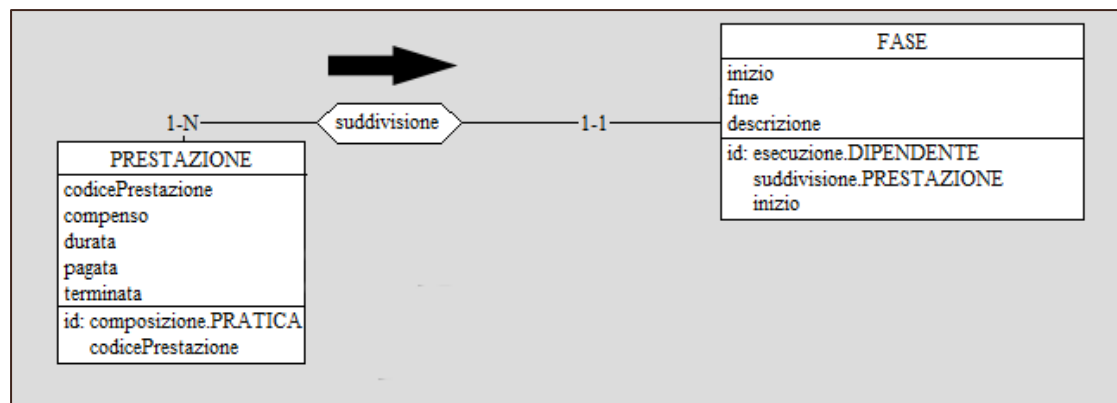
TOTALE	35000	*1/mese	35000/mese
--------	-------	---------	------------



V10 - Visualizzare le prestazioni la cui durata supera la durata media delle prestazioni

CONCETTO	COSTRUTTO	ACCESSO	TIPO
PRESTAZIONE	E	7000	L
suddivisione	A	21000	L
FASE	E	21000	L

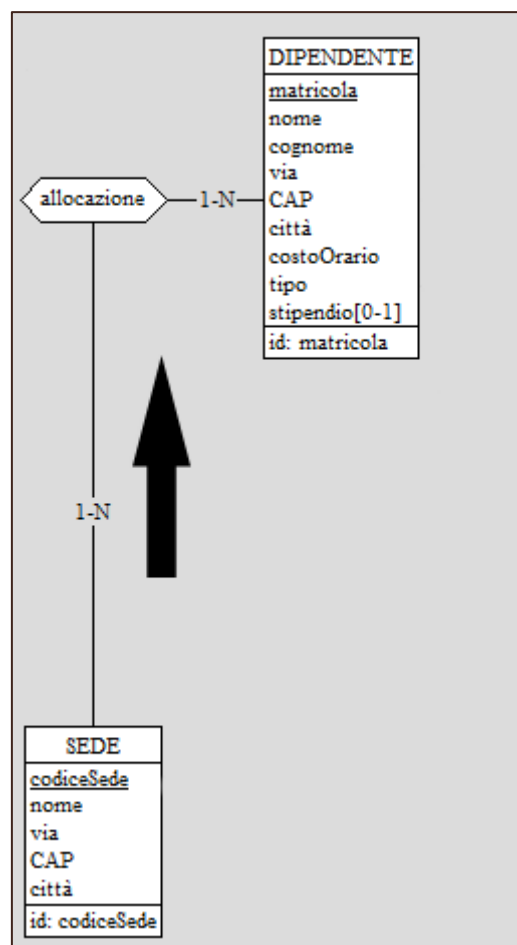
TOTALE	49000	*1/mese	49000/mese
--------	-------	---------	------------



V11 - Visualizzare i dipendenti e le relative allocazioni

CONCETTO	COSTRUTTO	ACCESSO	TIPO
SEDE	E	2	L
allocazione	A	20	L
DIPENDENTE	E	20	L

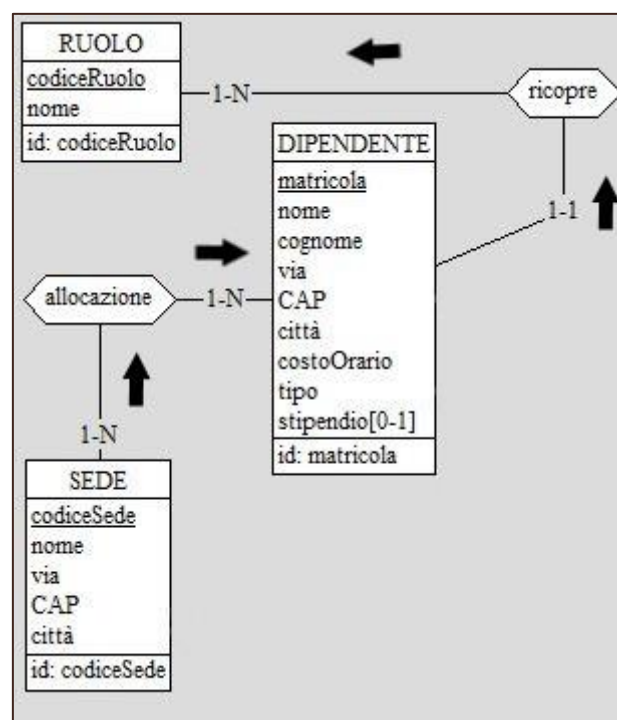
TOTALE	42	*1/mese	42/mese
--------	----	---------	---------



V12 - Visualizzare per una certa sede i dipendenti ed il relativo ruolo

CONCETTO	TIPO	ACCESSI	OP
SEDE	E	1	L
allocazione	A	10	L
DIPENDENTE	E	10	L
ricopre	A	10	L
RUOLO	E	10	L

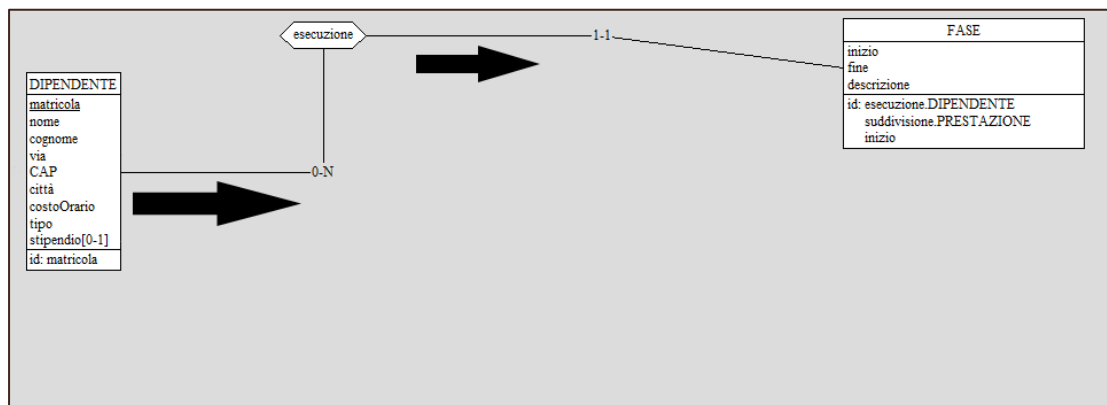
TOTALE	41	*10/mese	41/mese
--------	----	----------	---------



V14 - Visualizzare lo stipendio di un dipendente

CONCETTO	COSTRUTTO	ACCESSO	TIPO
DIPENDENTE (impiegato)	E	15	L
DIPENDENTE (collaboratore)	E	5	L
esecuzione	A	4200	L
FASE	E	4200	L

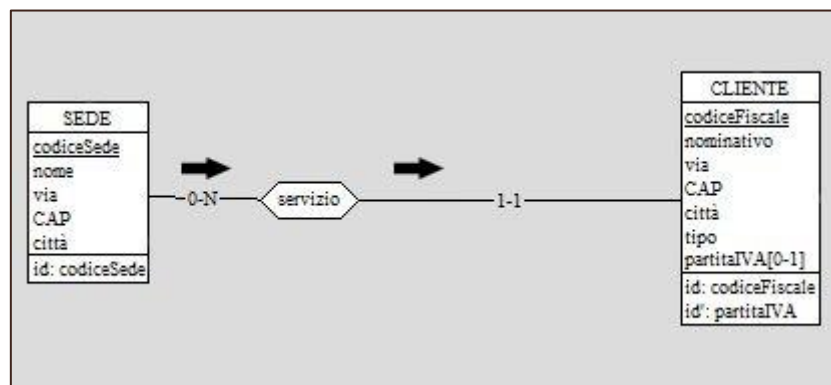
TOTALE	8420	*1/mese	8420/mese
--------	------	---------	-----------



V15 - Visualizzare per ogni sede le città dei relativi clienti

CONCETTO	TIPO	ACCESSI	OP
SEDE	E	2	L
servizio	A	100	L
CLIENTE	E	100	L

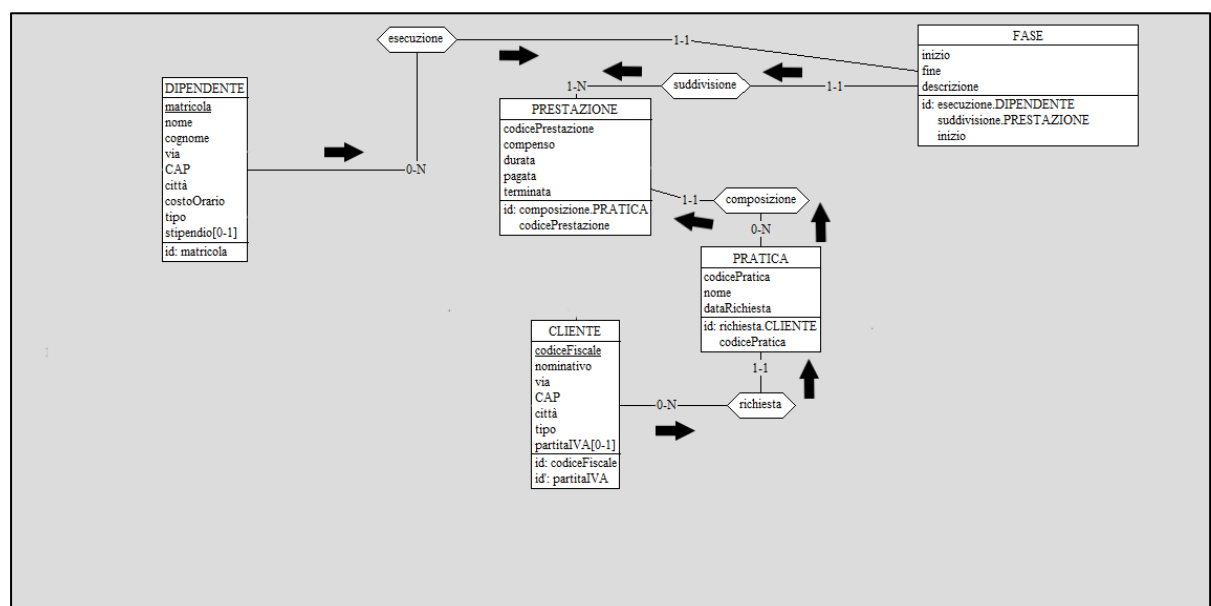
TOTALE	202	*1/mese	202/mese
--------	-----	---------	----------



V16 - Visualizzare le prestazioni per un cliente eseguite da un determinato dipendente

CONCETTO	COSTRUTTO	ACCESSO	TIPO
DIPENDENTE	E	1	L
esecuzione	A	1050	L
FASE	E	1050	L
suddivisione	A	1050	L
CLIENTE	E	1	L
richiesta	A	15	L
PRATICA	E	15	L
composizione	A	70	L
PRESTAZIONE	E	2100	L

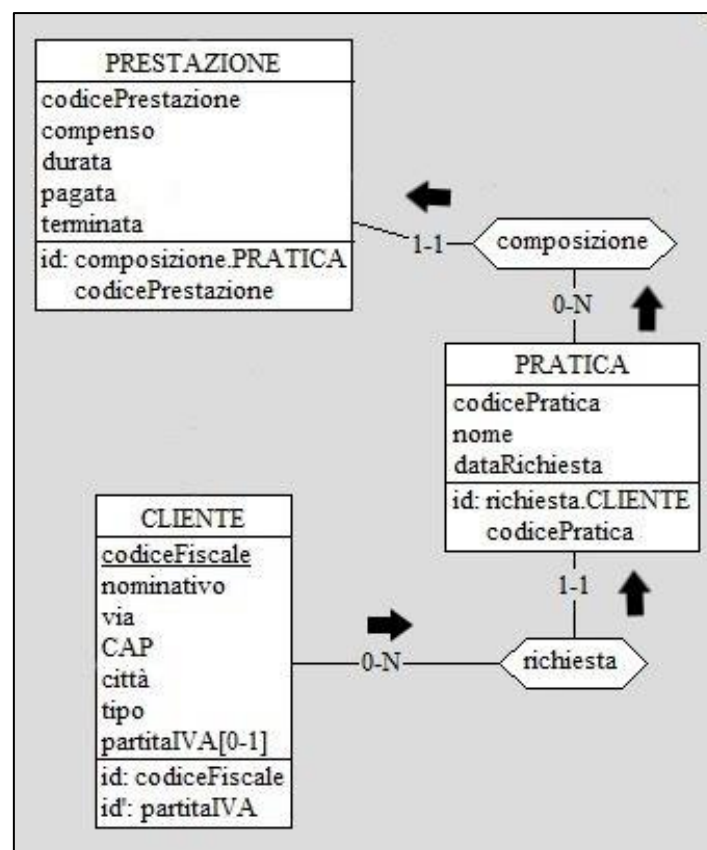
TOTALE	5352	*1/mese	5352/mese
--------	------	---------	-----------



V18 - Visualizzare il numero di prestazioni pagate per cliente in ordine decrescente

CONCETTO	COSTRUTTO	ACCESSO	TIPO
CLIENTE	E	100	L
richiesta	A	1500	L
PRATICA	E	1500	L
composizione	A	7000	L
PRESTAZIONE	E	7000	L

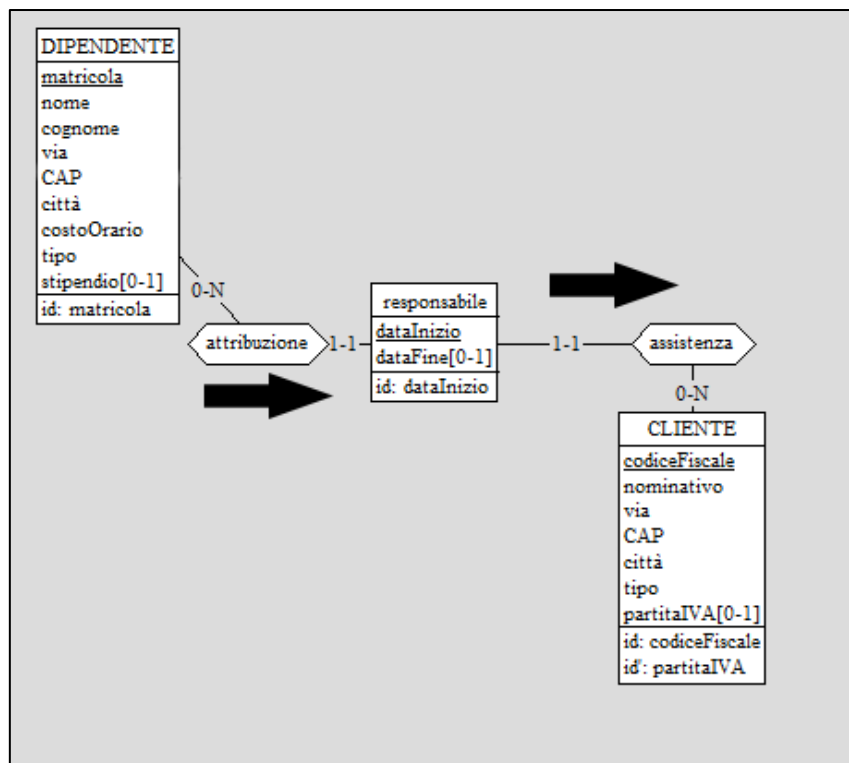
TOTALE	17100	*4/mese	68400/mese
--------	-------	---------	------------





V19 - Visualizzare i dipendenti e le relative responsabilità nel tempo, visualizzando anche chi non le ha mai avute

CONCETTO	COSTRUTTO	ACCESSO	TIPO
DIPENDENTE	E	20	L
responsabile	A	100	L
CLIENTE	E	100	L
TOTALE	220	*1/mese	220/mese



#### 4.4 – Analisi delle ridondanze

Per l'analisi delle ridondanze si è pensato di prendere in considerazione l'attributo "durata" presente in *prestazione*, poiché è una richiesta molto comune all'interno delle operazioni, in particolare la visualizzazione 10, 14 e 23. Infatti, senza la ridondanza, lo studio della durata di una prestazione avverrebbe accedendo ad ogni entità fase richiesta e facendo la differenza tra l'ora di inizio e di fine, entrambi suoi attributi, e, per ognuno dei risultati ottenuti, sommarli insieme; un'operazione un po' troppo complessa, no?

Il suddetto attributo infatti porterebbe alla risoluzione del problema, di seguito sono presenti i calcoli e le considerazioni da noi realizzate.

INSERIMENTO DI UNA FASE (frequenza 10/Giorno):

**Senza ridondanza:**

CONCETTO	COSTRUTTO	ACCESSO	TIPO
PRESTAZIONE	E	1	L
suddivisione	A	1	S
FASE	E	1	S
esecuzione	A	1	S
DIPENDENTE	E	1	L

Totale: 8

Frequenza totale:  $10 \times 8 \times \text{dipendenti (20)} = 1600/\text{Giorno}$ .

**Con ridondanza:**

CONCETTO	COSTRUTTO	ACCESSO	TIPO
PRESTAZIONE	E	1	L+S
suddivisione	A	1	S
FASE	E	1	S
esecuzione	A	1	S
DIPENDENTE	E	1	L

Totale: 10

Frequenza totale:  $10 \times 10 \times \text{dipendenti} = 2000/\text{Giorno}$ .

AGGIORNAMENTO DI UNA FASE (Frequenza 1/Mese):

**Senza ridondanza:**

CONCETTO	COSTRUTTO	ACCESSO	TIPO
FASE	E	1	L+S

Totale: 3

Frequenza totale:  $3 \times 1 \times \text{dipendenti} = 60/\text{Mese}$ .

**Con ridondanza:**

CONCETTO	COSTRUTTO	ACCESSO	TIPO
FASE	E	1	L+S
suddivisione	A	1	S
PRESTAZIONE	E	1	L+S

Totale: 7

Frequenza totale:  $7 \times \text{dipendenti} = 140/\text{Mese}$ .

VISUALIZZARE IL TOTALE DELLE ORE DEDICATE AD UN CLIENTE  
(frequenza 4/Mese):

**Senza ridondanza:**

CONCETTO	COSTRUTTO	ACCESSO	TIPO
CLIENTE	E	1	L
richiesta	A	$1500/100 = 15$	L
PRATICA	E	15	L
Composizione	A	$15*(7000/1500) = 75$	L
PRESTAZIONE	E	75	L
suddivisione	A	$75*(21000/7000) = 225$	L
FASE	E	225	L

Totale: 631

Frequenza totale:

$631*4 = 2524/\text{Mese}$ .

**Con ridondanza:**

CONCETTO	COSTRUTTO	ACCESSO	TIPO
CLIENTE	E	1	L
richiesta	A	$1500/100 = 15$	L
PRATICA	E	15	L
composizione	A	$15*7000/1500 = 70$	L
PRESTAZIONE	E	70	L

Totale: 171

Frequenza totale:  $171*4 = 684/\text{Mese}$ .

VISUALIZZARE LE PRESTAZIONI LA CUI DURATA SUPERA LA DURATA MEDIA DELLE  
PRESTAZIONI  
(frequenza 1/Mese)

**Senza ridondanza:**

CONCETTO	COSTRUTTO	ACCESSO	TIPO
PRESTAZIONE	E	7000	L
suddivisione	A	21000	L
FASE	E	21000	L

Totale:49000

Frequenza totale:49000/Mese.

**Con ridondanza:**

CONCETTO	COSTRUTTO	ACCESSO	TIPO
PRESTAZIONE	E	7000	L

Frequenza: 7000/Mese.

VISUALIZZARE LA DURATA COMPLESSIVA DI UNA PRESTAZIONE  
(frequenza 4/Mese):

**Senza ridondanza:**

CONCETTO	COSTRUTTO	ACCESSO	TIPO
PRESTAZIONE	E	1	L
suddivisione	A	$21000/7000 = 3$	L
FASE	E	3	L

Totale: 7

Frequenza totale:  $4 * 7 = 28/\text{Mese}$ .

**Con ridondanza:**

CONCETTO	COSTRUTTO	ACCESSO	TIPO
PRESTAZIONE	E	1	L

Frequenza totale:  $4 * 1 = 4/\text{Mese}$ .

**CALCOLO MENSILE FINALE:**

-con ridondanza=99612

-senza ridondanza=67828

Nonostante il costo esoso nell'inserimento e nell'aggiornamento, l'attributo ridondante "**durata**" nell'entità PRESTAZIONE risulta essere più vantaggioso guardando al totale delle operazioni che considerano la valutazione della durata complessiva di una o più prestazioni, a fronte di uno spreco esiguo di memoria.

Un altro attributo interessante da studiare per il calcolo delle ridondanze era la presenza o meno dell'attributo "**compenso**" all'interno dell'entità PRATICA, ma poiché utilizziamo una sola operazione che sarebbe influenzata dalla presenza di questo attributo e, anche se quest'ultima potrebbe giovarne, avremmo comunque dei costi eccessivi in caso di inserimento o aggiornamento; abbiamo quindi deciso di non utilizzarla, anche per un risparmio, seppur irrisorio, di memoria.

## **4.5 – Raffinamento dello schema**

- *Eliminazione delle gerarchie*

Nello schema principale sono presenti quattro differenti gerarchie, in tutte è stato deciso di effettuare un collasso verso l'alto, di seguito sono spiegati i motivi:

### DIPENDENTE (t, e)

Guadagna l'attributo "tipo" e l'attributo opzionale "stipendio".

### COSTO\_STRUTTURA (t, e)

Guadagna l'attributo "tipo" e l'attributo opzionale "quantità".

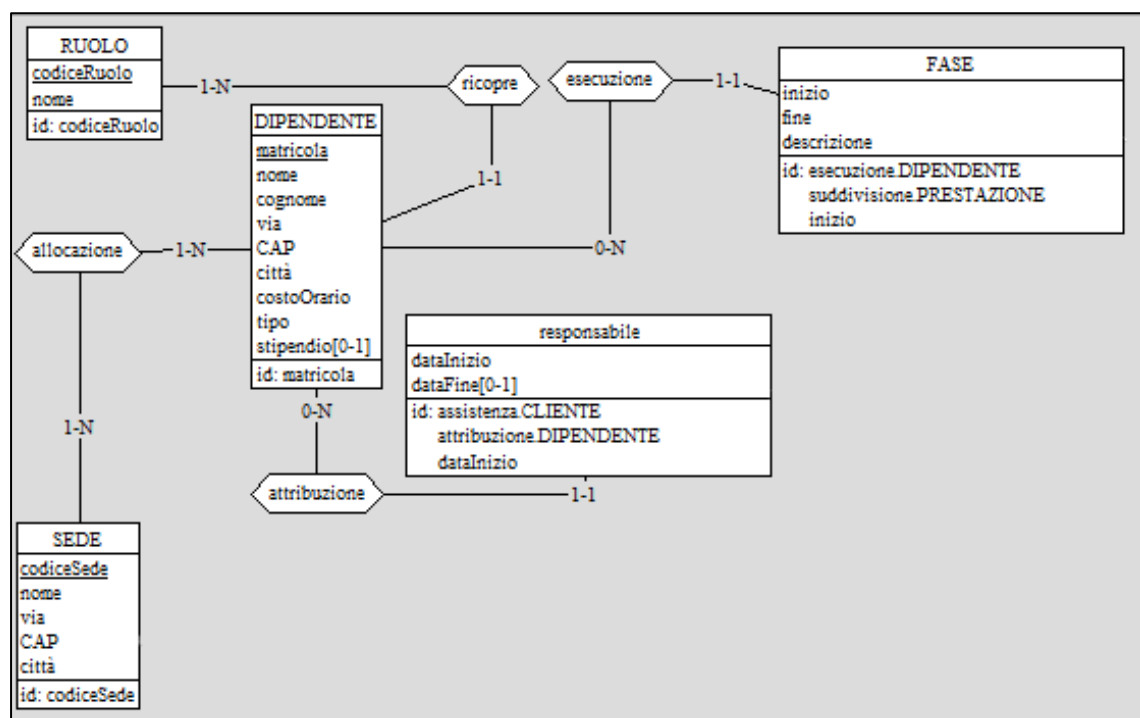
### CLIENTE (t, e)

Guadagna l'attributo "tipo" e l'attributo opzionale, che funge anche da chiave candidata, "partitaIVA".

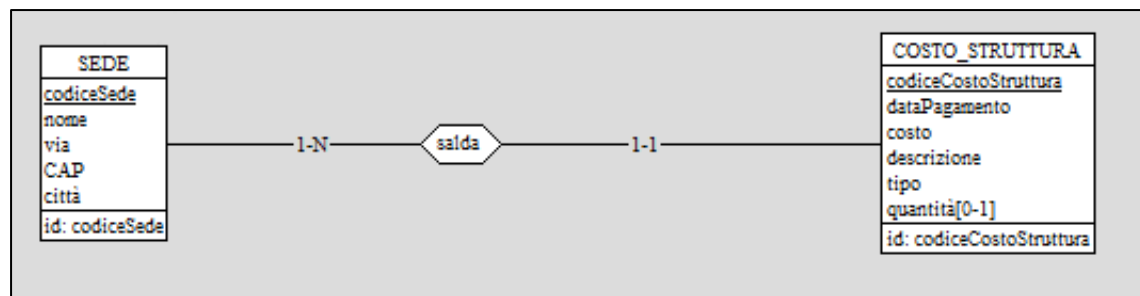
### SOTTOCATEGORIA (t, s)

Guadagna gli attributi "tipoPF" e "tipoAZ".

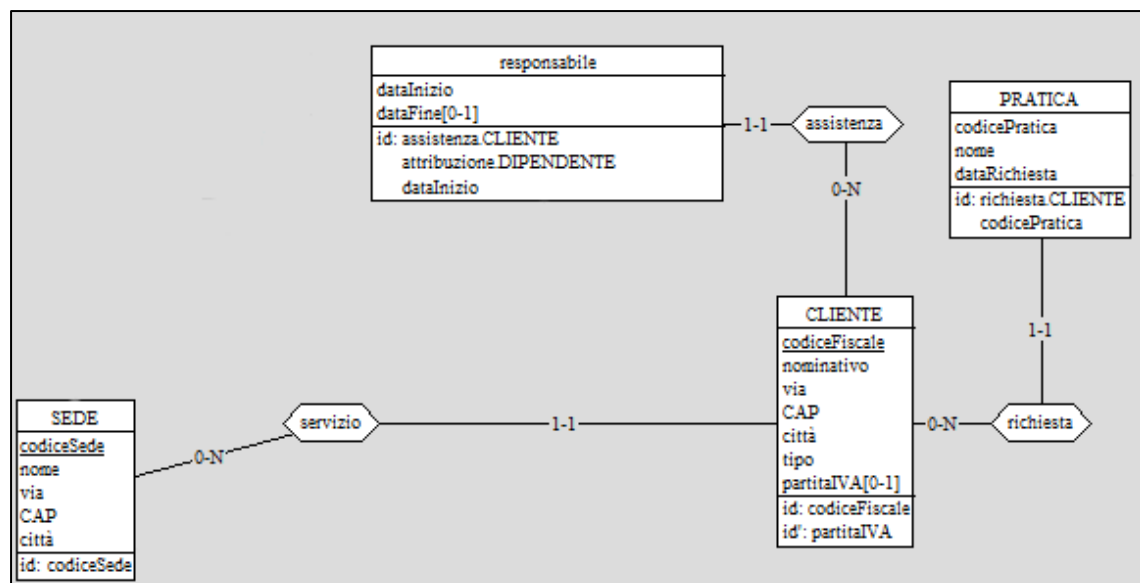
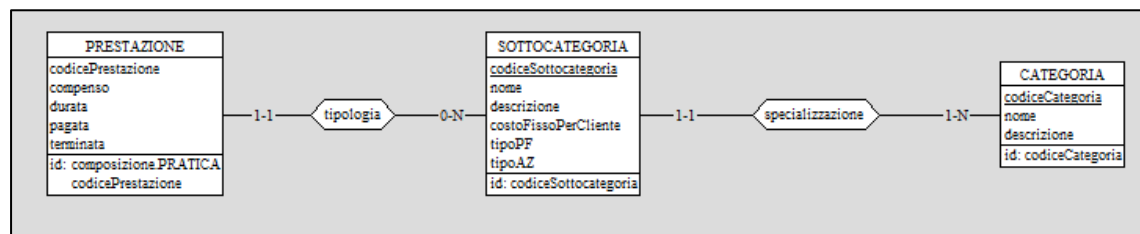
In DIPENDENTE, nonostante debba essere aggiunto l'attributo "tipo", per identificare l'impiegato o il collaboratore, e l'attributo opzionale "stipendio", date le numerose relazioni di tipo o-N o 1-N si preferisce avere un'unica tabella contenente tutti i dipendenti, anche perché le operazioni che richiedono l'accesso alle entità figlie sono spesso contigue.



Motivazioni simili valgono anche per **COSTO\_STRUTTURA** che, dopo il collasso, vedono l'acquisizione dell'attributo che era precedentemente presente nell'entità figlia MATERIALE. Inoltre, come nel caso precedente, tutte le operazioni che richiedono i costi di struttura lavorano contestualmente sulle entità figlie rendendo quindi inutile generare due differenti entità.



Le stesse motivazioni possono essere mosse per giustificare **CLIENTE** e **SOTTOCATEGORIA**, il quale non guadagna nessun attributo oltre ai selettori (“tipoPF” e “tipoAZ”).



- **Eliminazione degli attributi composti**

Nello schema relazionale è possibile notare la presenza di attributi composti, ovvero l'attributo “indirizzo” presente sia nell'entità DIPENDENTE che nell'entità CLIENTE; si procederà quindi alla scomposizione dell'attributo nei suoi tre sotto-attributi: “via”, “CAP” e “città”.

- **Scelta delle chiavi primarie**

È possibile notare le chiavi primarie già evidenziate nello schema E/R riportato nel paragrafo 3.3, data la natura del sistema informativo realizzato, molte entità vengono identificate da un campo numerico incrementale o da un codice identificativo (come il codice fiscale per i clienti o la matricola per i dipendenti) ma si possono muovere alcune osservazioni:

In primis nell'entità **CLIENTE** l'attributo "partita IVA", seppur opzionale, diviene una chiave candidata per i clienti di tipo "azienda" (vedi sopra).

Riguardo a **responsabile**, nata come un'associazione tra dipendente e cliente, descritta da due attributi quali "dataInizio" e "dataFine", si è deciso, durante la progettazione, così il successivo lavoro di traduzione logica dello schema, di reificarla in un'entità vera e propria, poiché la relazione che si veniva a creare era N-M; questa entità (e analogamente anche l'entità **FASE**) utilizza come chiave primaria l'attributo "dataInizio", importando le chiavi esterne dalle altre due entità che mette in associazione, ma questo verrà analizzato nel paragrafo successivo.

- **Chiavi esterne**

Nello schema realizzato si fa largo uso di foreign key, in particolare è possibile notare che il "percorso" che si viene a costruire da cliente a fase si concretizza tramite l'utilizzo di chiavi esterne "a cascata". Di seguito l'analisi delle singole entità:

**PRATICA** viene identificata dal suo "codicePratica" e, tramite la relazione *richiesta*, dal "codiceFiscale" importato dal **CLIENTE** che la richiede.

**PRESTAZIONE** viene identificata dal suo "codicePrestazione" e, tramite la relazione *composizione*, dalle FK importate, quindi "codicePratica" di **PRATICA** e "codiceFiscale" del **CLIENTE** richiedente.

**FASE** viene identificata dall'attributo "inizio" di tipo 'DateTime' e, tramite la relazione *suddivisione ed esecuzione*, dalle FK importate, quindi "codicePrestazione" della prestazione della quale fa parte, "codicePratica", della pratica composta dalla suddetta prestazione, "codiceFiscale" del cliente che ha richiesto la precedente pratica e dalla "matricola" del dipendente che esegue la fase in questione.

**Responsabile** viene identificato da "dataInizio" e dalle chiavi importate da *attribuzione* e *assistenza*, quindi da "matricola" di **DIPENDENTE** e da "codiceFiscale" di **CLIENTE**.

#### ***4.6 – Traduzione delle entità e associazioni in relazioni***

**allocazione**(matricola, codiceSede)

FK: codiceSede

FK: matricola

**CATEGORIA**(codiceCategoria, nome, descrizione)

**CLIENTE**(codiceFiscale, nominativo, partitaIVA[0-1], codiceSede)

FK: codiceSede

**COSTO\_STRUTTURA**(codiceCostoStruttura, dataPagamento, costo, descrizione, tipo, quantità[0-1], codiceSede)

FK: codiceSede

**DIPENDENTE**(matricola, nome, cognome, costoOrario, stipendio [0-1], codiceRuolo)

FK: codiceRuolo

**FASE**(codiceFiscale, codicePratica, codicePrestazione, matricola, inizio, fine, descrizione)

FK: codiceFiscale, codicePratica, codicePrestazione (REFERENCES

**PRESTAZIONE**)

FK: matricola

**PRATICA**(codiceFiscale, codicePratica, nome, dataRichiesta)

FK: codiceFiscale

**PRESTAZIONE**(codiceFiscale, codicePratica, codicePrestazione, compenso, durata, pagata[0-1], terminata[0-1], codiceSottocategoria)

FK: codiceSottocategoria

FK: codiceFiscale, codicePratica (REFERENCES **PRATICA**)

**RUOLO**(codiceRuolo, nome)

**SEDE**(codiceSede, nome, via, CAP, città)

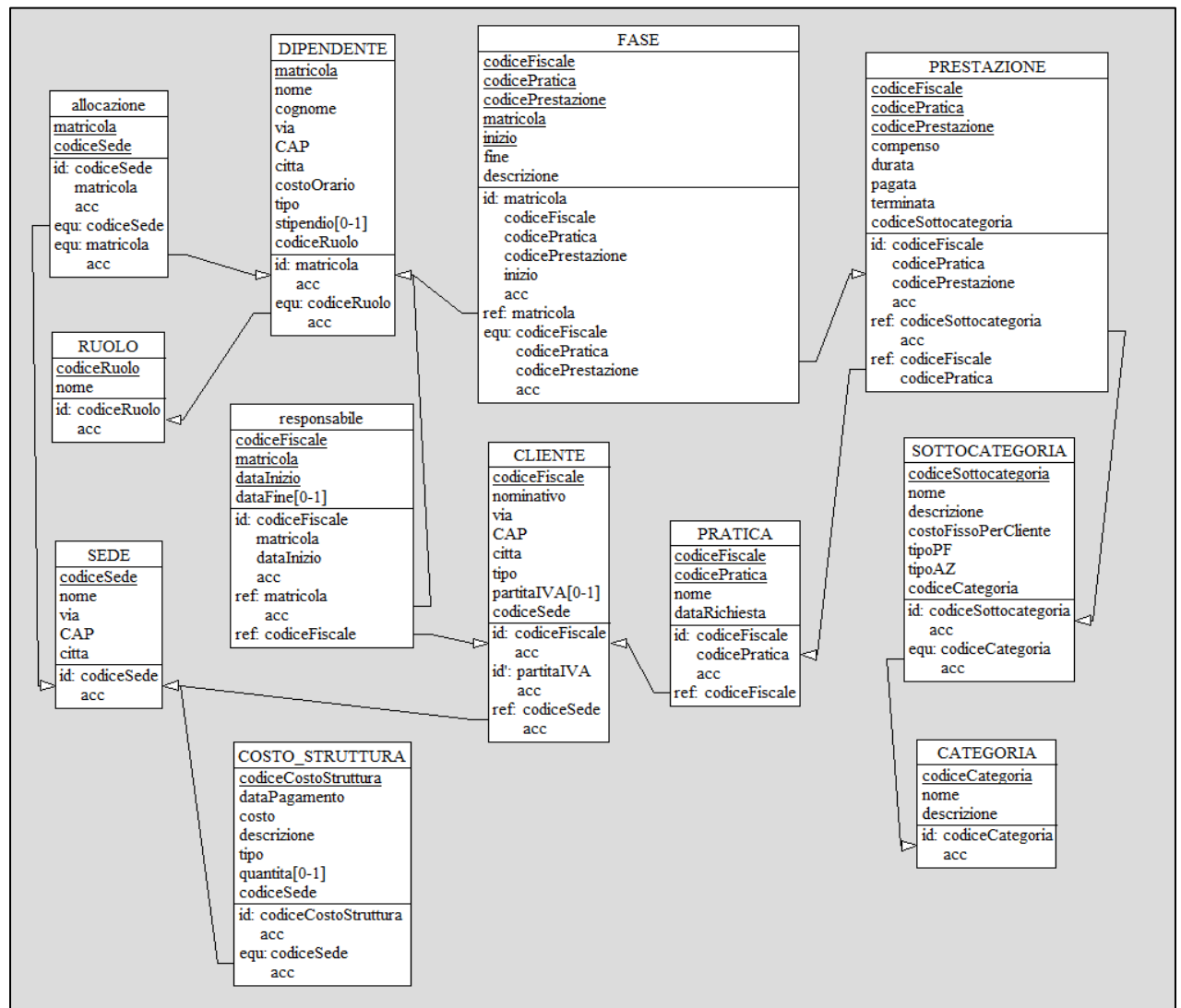
**SOTTOCATEGORIA**(codiceSottocategoria, nome descrizione, costoFissoPerCliente, tipoPF, tipoAZ, codiceCategoria)

FK: codiceCategoria



## 4.7 – Schema logico finale

Di seguito è riportato lo schema logico finale:



## 4.8 – Stesura delle principali operazioni in linguaggio SQL

- Visualizzare tutte le fasi svolte per un determinato cliente e info su di esso, sulla prestazione e sulla pratica
- ```

SELECT
c.codiceFiscale,c.nominativo,c.tipo,c.partitaIVA,pa.codicePratica,pa.nome,pe.codicePrestazione,pe.compenso AS
compensoPrestazione,pe.pagata,f.descrizione AS
descrizioneFase,f.inizio AS inizioFase,f.fine AS
fineFase,f.Matricola
FROM cliente c,praticapa,prestazione pe,fase f
WHERE c.codiceFiscale = "codiceFiscale"
  
```

```

AND c.codiceFiscale = pa.codiceFiscale
AND pa.codiceFiscale = pe.codiceFiscale AND pa.codicePratica =
pe.codicePratica
AND pe.codiceFiscale = f.codiceFiscale AND pe.codicePratica =
f.codicePratica AND pe.codicePrestazione = f.codicePrestazione
ORDER BY pa.CodicePratica,pe.CodicePrestazione

```

- **Visualizzare compensi e dettagli di una certa pratica richiesta da un cliente**

```

SELECT codiceFiscale, codicePrestazione,
compenso, (compenso+s.CostoFissoPerCliente) as compensoTotale,
pagata, terminata
FROM PRESTAZIONE prest, SOTTOCATEGORIA s
WHERE codicePratica="codicePratica"
AND codiceFiscale="codiceFiscale"
AND prest.codiceSottocategoria = s.codiceSottocategoria

```

- **Visualizzare tutte le prestazioni terminate e/o pagate in base a un determinato cliente**

```

SELECT c.CodiceFiscale, c.Nominativo, c.Tipo, c.PartitaIVA, pa.Codi
cePratica, pa.Nome, pe.CodicePrestazione, pe.Compenso AS
compensoPrestazione, pe.Pagata, pe.Terminata
FROM cliente c, pratica pa, prestazione pe
WHERE .codiceFiscale = "codiceFiscale"
AND pe.pagata = 0/1
AND pe.terminata = 0/1
AND c.codiceFiscale = pa.codiceFiscale

AND pa.codiceFiscale = pe.codiceFiscale AND pa.codicePratica =
pe.codicePratica
ORDER BY pa.CodicePratica, pe.CodicePrestazione

```

- **Visualizzare per ogni cliente il relativo storico dei responsabili**

```

SELECT
c.codiceFiscale, c.nominativo, d.matricola, d.nome, d.cognome, r.da
taInizio, r.dataFine
FROM CLIENTE c, responsabile r, DIPENDENTE d
WHERE c.codiceFiscale=r.codiceFiscale
AND d.matricola=r.matricola
GROUP BY
c.codiceFiscale, c.nominativo, d.matricola, d.nome, d.cognome, r.da
taInizio, r.dataFine

```

ORDER BY c.codiceFiscale,r.dataInizio,r.dataFine

- **Visualizzare per ogni sede i relativi costi struttura**

```
SELECT
s.codiceSede,s.nome,cs.dataPagamento,cs.costo,cs.descrizione,c
s.quantita,cs.tipo
FROM SEDE s,COSTO_STRUTTURA cs
WHERE cs.codiceSede=s.codiceSede
GROUP BY
s.codiceSede,s.nome,cs.dataPagamento,cs.costo,cs.descrizione,c
s.quantita,cs.tipo
```

- **Visualizzare il totale delle ore dedicate ad un cliente**

```
SELECT c.codiceFiscale,c.nominativo,c.tipo,(SELECT
SEC_TO_TIME(SUM(TIME_TO_SEC(durata)))) AS oreDedicate
FROM cliente c,prestazione pe
WHERE c.codiceFiscale = "codicefiscale"
AND c.codiceFiscale = pe.codiceFiscale
```

- **Visualizzare dei 5 dipendenti che hanno fatto meno fasi**

```
SELECT d.matricola, d.nome, d.cognome, COUNT(*) asnumeroFasi
FROM DIPENDENTE d, FASE f
WHERE f.matricola = d.matricola
GROUP BY d.matricola, d.nome, d.cognome
ORDER BY COUNT(*) ASC,d.cognome,d.nome
LIMIT 5
```

- **Visualizzare la sottocategoria (e la relativa categoria) più richiesta**

```
SELECT S.nome, C.nome, COUNT(P.codiceSottocategoria)
asNumeroAttivita
FROM PRESTAZIONE p, SOTTOCATEGORIA s, CATEGORIA c
WHERE p.codiceSottocategoria = s.codiceSottocategoria
AND s.codiceCategoria = c.codiceCategoria
GROUP BY p.codiceSottocategoria, s.nome, C.nome
ORDER BY COUNT(p.codiceSottocategoria) DESC
```

- **Visualizzare le prestazioni la cui durata supera la durata media delle prestazioni**

```
SELECT codiceFiscale,codicePratica,codicePrestazione, durata
FROM PRESTAZIONE
WHERE TIME_TO_SEC(durata) > (SELECTAVG(TIME_TO_SEC(durata))
from PRESTAZIONE)
GROUP BY codiceFiscale,codicePratica,codicePrestazione,durata
```

- **Visualizzare i dipendenti e le relative allocazioni**

```
SELECT D.matricola,D.nome, D.cognome, S.nome
FROM DIPENDENTE D, SEDE S, allocazione A
WHERE D.matricola = A.matricola
AND A.codiceSede = S.codiceSede
```

- **Visualizzare per una certa sede i dipendenti ed il relativo ruolo**

```
SELECT D.matricola,D.nome, D.cognome, S.nome
FROM DIPENDENTE d, SEDE s, allocazione a
WHERE d.matricola = a.matricola
AND a.codiceSede = s.codiceSede
```

- **Visualizzare l'elenco dei clienti di una certa tipologia**

- **Persona fisica:**

```
SELECT *
FROM CLIENTE
WHERE tipo = "persona_fisica"
```

- **Azienda**

```
SELECT *
FROM CLIENTE
WHERE tipo = "azienda"
```

- **Visualizzare lo stipendio di un dipendente**

Per questa query abbiamo deciso di fare due operazioni separate per gli impiegati e per i collaboratori ed in seguito unirle

```
SELECT d.matricola, nome, cognome, stipendio
FROM DIPENDENTE d
WHERE tipo IN ('I')
```

```
SELECT DISTINCT d.matricola, nome,cognome,
(costoOrario/60)*(SELECT(SUM(TIMESTAMPDIFF(MINUTE,inizio,fine)
))
FROM DIPENDENTE D1, FASE F1
WHERE D.matricola=D1.matricola
AND F.matricola=F1.matricola
AND F1.inizio >= 'datainizio'
AND F1.fine <= 'datafine') AS stipendio
FROM dipendente D, fase F
WHERE tipo IN ('C')
AND F.matricola=D.matricola
```

In alternativa si poteva utilizzare l'espressione CASE, ma non è stata affrontata durante il corso.

- **Visualizzare per ogni sede le città dei relativi clienti**

```
SELECT s.codiceSede,s.nome,c.nominativo,c.citta
FROM sede s,cliente c
WHERE c.codiceSede = s.codiceSede
group by s.codiceSede,s.nome,c.nominativo,c.citta
```

- **Visualizzare le prestazioni per un cliente eseguite da un determinato dipendente**

```
SELECT DISTINCT
matricola,PA.nomeasNomePratica,PE.codicePrestazione,S.codiceSo
ttocategoria,compenso,(compenso+s.costofissoPerCliente)AS
compensoTotale,durata,pagata,terminata
FROM FASE F, PRESTAZIONE PE, PRATICA PA,SOTTOCATEGORIA s
WHERE F.codicePrestazione = PE.codicePrestazione
AND F.codicePratica = PE.codicePratica
AND F.codiceFiscale = PE.codiceFiscale
AND PE.codicePratica = PA.codicePratica
AND PE.codiceFiscale = PA.codiceFiscale
AND PE.codiceSottocategoria=S.codiceSottocategoria
AND PA.codiceFiscale = "codiceFiscale"
AND F.matricola = "matricola"
```

- **Visualizzare il numero di prestazioni pagate per cliente in ordine decrescente**

```
SELECT C.nominativo, COUNT(*) AS numPrestazioniPagate
FROM PRESTAZIONE P, CLIENTE C
WHERE P.codiceFiscale = C.codiceFiscale
AND pagata="true"
GROUP BY C.codiceFiscale
ORDER BY COUNT(*) DESC
```

- **Visualizzare la sottocategoria (e la relativa categoria) più richiesta.**

```
SELECT S.nome, C.nome, COUNT(P.codiceSottocategoria)
asNumeroAttivita
FROM PRESTAZIONE P, SOTTOCATEGORIA S, CATEGORIA C
WHERE P.codiceSottocategoria = S.codiceSottocategoria
AND S.codiceCategoria = C.codiceCategoria
GROUP BY P.codiceSottocategoria, S.nome, C.nome
ORDER BY COUNT(P.codiceSottocategoria) DESC
```

- **Visualizzare i dipendenti e le relative responsabilità nel tempo, visualizzando anche chi non le ha mai avute**

```
SELECT d.matricola, d.nome, d.cognome, d.tipo, c.codiceFiscale,
c.nominativo, r.dataInizio, r.dataFine
FROM dipendente d
LEFT JOIN responsabile AS r on d.matricola = r.matricola
LEFT JOIN cliente AS c on r.codiceFiscale = c.codiceFiscale
GROUP BY d.matricola, d.nome, d.cognome, d.tipo,
c.codiceFiscale, c.nominativo, r.dataInizio, r.dataFine
ORDER BY d.matricola, r.dataIniziodesc
```

#### ***4.9 – Inserimento di vincoli generici***

Per la realizzazione completa del database si è deciso di aggiungere dei vincoli di tupla arbitrari mediante l'uso della clausola 'CHECK'; in questo modo, in caso venisse richiesta da applicativo un'azione non consentita, verrà generato un errore, ciò permetterà dunque di preservare l'integrità del database.

Di seguito sono riportati i vincoli realizzati, insieme ad una breve descrizione:

- `ALTER TABLE cliente ADD CONSTRAINT typeCheckCliente CHECK (tipo in ('P', 'A'));`

Controlla che l'attributo "tipo" del CLIENTE sia 'A' o 'P' (azienda o persona fisica).

- `ALTER TABLE costo_struttura ADD CONSTRAINT typeCheckCostoStruttura CHECK (tipo in ('I', 'P', 'M'));`

Controlla che l'attributo "tipo" di COSTO\_STRUTTURA sia 'I', 'P' o 'M' (Imprevisto, periodico o materiale).

- `ALTER TABLE costo_struttura ADD CONSTRAINT typeCostoCostoStruttura CHECK (costo >= 0);`

Controlla che venga inserito un COSTO\_STRUTTURA che abbia un valore per il "costo", ovvero per l'importo totale, maggiore o uguale a 0.

- ALTER TABLE costo\_struttura ADD CONSTRAINT  
checkQuantitaCostoStruttura CHECK ((tipo IN ('M') AND  
quantita >= 1 ) OR (tipo IN ('P','I') AND quantita IS NULL ));

Si assicura che l'attributo "quantità" dei costi struttura di tipo "materiale" sia presente solo se il tipo di COSTO\_STRUTTURA è effettivamente 'M' (materiale), altrimenti deve essere NULL.

- ALTER TABLE dipendente ADD CONSTRAINT  
checkCostoOrarioDipendente CHECK (costoOrario >= 0);

Si assicura che un DIPENDENTE abbia un costoOrario maggiore o uguale a 0.

- ALTER TABLE dipendente ADD CONSTRAINT checkTipoDipendente  
CHECK (tipo in ('I','C'));

Controlla che l'attributo "tipo" in DIPENDENTE sia 'I' o 'C' (impiegato o collaboratore)

- ALTER TABLE dipendente ADD CONSTRAINT  
checkStipendioDipendente CHECK ((tipo IN('I') AND stipendio  
>= 0) OR (tipo IN('C') AND stipendio IS NULL));

Controlla che se un DIPENDENTE è di tipo 'I' (impiegato) allora lo stipendio deve essere maggiore o uguale a 0, altrimenti deve essere NULL.

- ALTER TABLE fase ADD CONSTRAINT checkDataFineFase CHECK  
(inizio <= fine);

Si assicura che il valore di inizio di una fase sia minore o uguale al suo valore di fine.

- ALTER TABLE prestazione ADD CONSTRAINT checkCompensoPrestazione  
CHECK (compenso >= 0);

Controlla che il compenso di una prestazione sia almeno 0

- ALTER TABLE responsabile ADD CONSTRAINT checkLegalitaDataFine CHECK  
(dataFine IS NULL OR dataInizio <= dataFine);

Controlla che in responsabile la data di fine rapporto sia NULL o, nel caso in cui sia presente, che l'inizio non avvenga dopo la data di fine

- `ALTER TABLE sottocategoria ADD CONSTRAINT checkCostoFissoSottocategoria CHECK (costoFissoPerCliente >= 0);`  
Controlla che il costo fisso legato ad una sottocategoria sia almeno 0

#### ***4.10 – Inserimento dei trigger***

L'inserimento dei trigger all'interno del database permette l'ottimizzazione del database stesso, controllando e automatizzando l'inserimento dei dati, consolidando e centralizzando la logica generalmente implementata nell'applicativo. Questa soluzione permetterà di risparmiare tempo e memoria, essendo i trigger delle procedure attivate da determinati eventi, come ad esempio l'inserimento di un record.

Di seguito sono riportati i trigger utilizzati all'interno del DBMS:

##### **4.10.1 – trigger per l'entità FASE**

- Aggiunge la differenza tra inizio e fine della fase aggiunta(nel campo durata di prestazione)

```
TRIGGER_FASE_DURATA_INSERTIMENTO(AFTER INSERT)
BEGIN
UPDATE prestazione SET
durata=ADDTIME(durata,TIMEDIFF(NEW.fine,NEW.inizio))
WHERE codiceFiscale=NEW.codiceFiscale
AND codicePrestazione=NEW.codicePrestazione
AND codicePratica=NEW.codicePratica;
END
```

- Modifica della durata di una prestazione.

poiché la fase, quindi i suoi campi inizio e data, è aggiornabile, bisogna modificare il valore di durata nella prestazione correlata.

la prima query di update serve a sottrarre la differenza tra inizio e fine con i valori di prima della modifica, in modo da riportare durata ad uno stato "come se quel cambiamento non fosse mai successo", poi vengono sommati valori aggiornati



```

TRIGGER_FASE_DURATA_AGGIORNAMENTO(AFTER UPDATE)
BEGIN
UPDATE prestazione SET
    durata=SUBTIME(durata,TIMEDIFF(OLD.fine,OLD.inizio)) WHERE
    codiceFiscale=OLD.codiceFiscale AND
    codicePrestazione=OLD.codicePrestazione AND
    codicePratica=OLD.codicePratica;

```

```

UPDATE prestazione SET
    durata=ADDTIME(durata,TIMEDIFF(NEW.fine,NEW.inizio)) WHERE
    codiceFiscale=NEW.codiceFiscale AND
    codicePrestazione=NEW.codicePrestazione AND
    codicePratica=NEW.codicePratica;
END

```

- Relazione tra sede del dipendente e del cliente ove viene richiesta una pratica  
se la seguente condizione non viene rispettata la query non viene eseguita: la sede scelta dal cliente deve essere tra le sedi del dipendente che svolge la fase per quel dipendente

```

TRIGGER_FASE_SEDE(BEFORE INSERT)
BEGIN
IF NOT EXISTS (SELECT * FROM cliente C WHERE C.codiceFiscale =
    new.codiceFiscale AND C.codiceSede IN (SELECT
    allocazione.codiceSede FROM allocazione WHERE
    allocazione.matricola = new.matricola)) THEN
    SIGNAL SQLSTATE '45001';
END IF;
END

```

In questo caso ad esempio se provassimo a fare un inserimento del genere:  
 insert into fase  
 (codiceFiscale,codicePratica,codicePrestazione,matricola,inizio,fine,descrizione)  
 VALUES('BNCGRG01S27I917O',2,6,2,'2011-12-18 13:17:17','2011-12-18 14:17:17','prova');  
 andremmo, giustamente, incontro ad un errore.

#### **4.10.2 – trigger per l'entità RESPONSABILE**

- Controlla che il nuovo responsabile sia un IMPIEGATO, e tra le sedi in cui lavora, ci sia anche quella in cui il cliente richiede il servizio

```

TRIGGER_RESPONSABILE_SEDE(BEFORE INSERT)
BEGIN
IF NOT EXISTS (SELECT * FROM cliente C WHERE C.codiceFiscale
               =new.codiceFiscale AND C.codiceSede IN (SELECT
               allocazione.codiceSede FROM allocazione,dipendente WHERE
               dipendente.matricola=allocazione.matricola AND
               dipendente.tipo='I' AND allocazione.matricola =
               new.matricola)) THEN
               SIGNAL SQLSTATE '45001';
END IF;
END

```

In questo caso ad esempio se provassimo a fare un inserimento del genere:

```

insert into responsabile (codiceFiscale,matricola,dataInizio,dataFine)
VALUES('BNCGRG01S27I917O',3,'2071-12-18','2011-12-18 14:17:17');

```

andremmo, nuovamente, incontro ad un errore.

#### 4.10.3 – trigger per l'entità PRATICA

Essendo impossibile applicare il campo AUTO\_INCREMENT a chiavi composte abbiamo optato per utilizzare un trigger per poter renderlo possibile

(BEFORE INSERT)

```

BEGIN SET NEW.codicePratica = (SELECT IFNULL(MAX(codicePratica),0) + 1
FROM pratica WHERE pratica.codiceFiscale = NEW.codiceFiscale); END

```

Questo trigger setta il codicePratica con l'ausilio della funzione IFNULL che accetta due parametri.

Questa funzione permette di restituire un parametro alternativo se l'espressione è NULL.

Nel caso in cui l'espressione non sia NULL allora restituisce il risultato dell'espressione.

MAX(codicePratica) restituisce il massimo valore di codicePratica, a cui verrà poi incrementato di 1 e così sarà settato il nuovo codicePratica.

Nel caso in cui non ci siano pratiche, MAX(codicePratica) restituirebbe NULL e quindi grazie all'ausilio della funzione in realtà restituirà 0 incrementato di 1 e quindi quella sarà la prima.

#### 4.10.4 – trigger per l'entità PRESTAZIONE

- **Controlla che la tipologia del cliente sia compatibile con la tipologia della sottocategoria della prestazione richiesta**

```
TRIGGER_PRESTAZIONE_SOTTOCATEGORIA(BEFORE INSERT)
BEGIN
DECLARE TIPOCLIENTE CHAR;
DECLARE TIPOAZ INT;
DECLARE TIPOPF INT;
SET TIPOCLIENTE = (SELECT cliente.tipo FROM cliente WHERE
    cliente.codiceFiscale=new.codiceFiscale);
SET TIPOAZ = (SELECT sottocategoria.tipoAZ from
    sottocategoria wheresottocategoria.codiceSottocategoria =
    new.codiceSottocategoria);
SET TIPOPF = (SELECT sottocategoria.tipoPF from
    sottocategoria wheresottocategoria.codiceSottocategoria =
    new.codiceSottocategoria);

    IF ((TIPOCLIENTE='A' AND TIPOAZ=0) OR (TIPOCLIENTE='P'
    AND TIPOPF=0)) THEN SIGNAL SQLSTATE '45001';
    END IF;
END
```

- **Essendo impossibile applicare il campo AUTO\_INCREMENT a chiavi composte abbiamo optato per utilizzare un trigger per poter renderlo possibile**

```
(BEFORE INSERT)
BEGIN

    SET NEW.codicePrestazione = (SELECT IFNULL(MAX(codicePrestazione),0) +
    1 FROM prestazione WHERE prestazione.codicePratica = NEW.codicePratica
    AND prestazione.codiceFiscale = NEW.codiceFiscale);

END
```

Questo trigger ha lo stesso comportamento di quello analizzato precedentemente nel 4.10.3 (trigger Pratica)

## 5 – Progettazione dell'applicazione

### 5.1 - Descrizione dell'architettura dell'applicazione realizzata

Si è deciso di sviluppare l'applicazione tramite il linguaggio C#.NET.

La tipologia di RDBMS che abbiamo voluto utilizzare è una fork di MySQL ovvero MariaDB, la motivazione per questa scelta è dovuta al fatto che MariaDB supporta le CONSTRAINT CHECK mentre MySQL no.

L'interazione tra l'applicativo e il database è stato fatto in due maniere:

- Linq2DB.MySql: è una libreria molto utile che permette di poter utilizzare la sintassi di Linq interfacciandosi con un database MySQL
- MySql.Data.MySqlClient: è la libreria classica che permette l'interazione fra l'applicativo e il database MySQL. (utilizzata solo per alcune specifiche query)

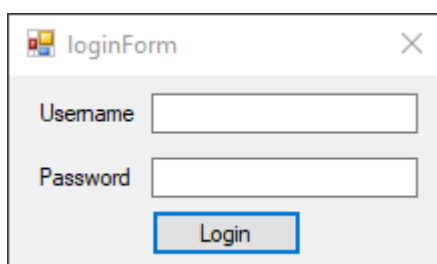
All'avvio dell'applicazione si presenterà una form di login.

Tramite inserimento di username e password si potrà accedere al resto dell'applicazione, questo è stato fatto perché certi utenti hanno dei privilegi di amministratore e altri no.

Nella tabella "account" (non menzionata negli schemi precedenti perché è ad uso dell'applicazione) sono contenute le informazioni relative all'username, password, matricola associata e un campo booleano per specificare se quell'account possiede o meno i privilegi di amministratore.

In questa maniera quando un utente riesce ad autenticarsi si sa chi di questi sia semplicemente un dipendente oppure sia un amministratore e di conseguenza si nascondono o rendono visibili alcune funzioni all'interno dell'applicazione.

#### 5.1.1 - Form di login:



The image shows a Windows-style dialog box titled "loginForm". It contains two text input fields: "Username" and "Password". Below these fields is a button labeled "Login". The "Login" button is highlighted with a blue border.

### 5.1.2 - Schermata principale:

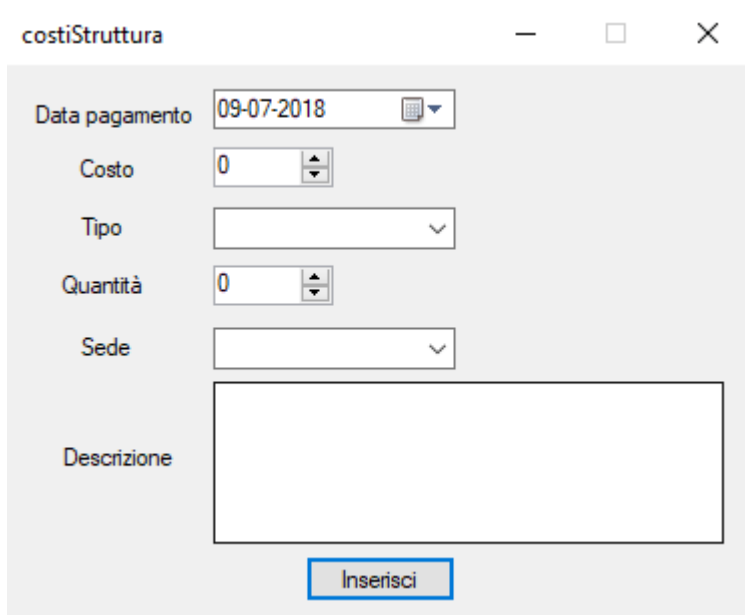
Questa è la form principale del programma utilizzata per l'inserimento di pratiche, prestazioni e fasi e la modifica di prestazioni e fasi.

Scegliendo il cliente interessato si possono vedere le sue pratiche tramite il bottone “visualizza” oppure inserirne una nuova tramite il bottone “inserisci”.

Una volta che si sceglie di vedere le pratiche di un cliente verranno mostrate nel DataGridView a destra dove tramite un semplice “click” su una pratica mostrerà a sua volta tutte le prestazioni di quella pratica selezionata e di conseguenza verranno anche compilati i campi di quella pratica rendendo anche accessibile la GroupBox Prestazione che contiene i campi per l'inserimento o la modifica di quest'ultima.

Questo metodo è poi riutilizzato anche per le prestazioni e le fasi, ovvero una volta che vengono mostrate le prestazioni cliccando su quella interessata se ne visualizzeranno tutte le sue fasi rendendo accessibile anche la GroupBox Fase.

### 5.1.3 - Schermata di inserimento costi struttura:



The screenshot shows a window titled "costiStruttura" with standard Windows window controls (minimize, maximize, close). The form contains the following fields:

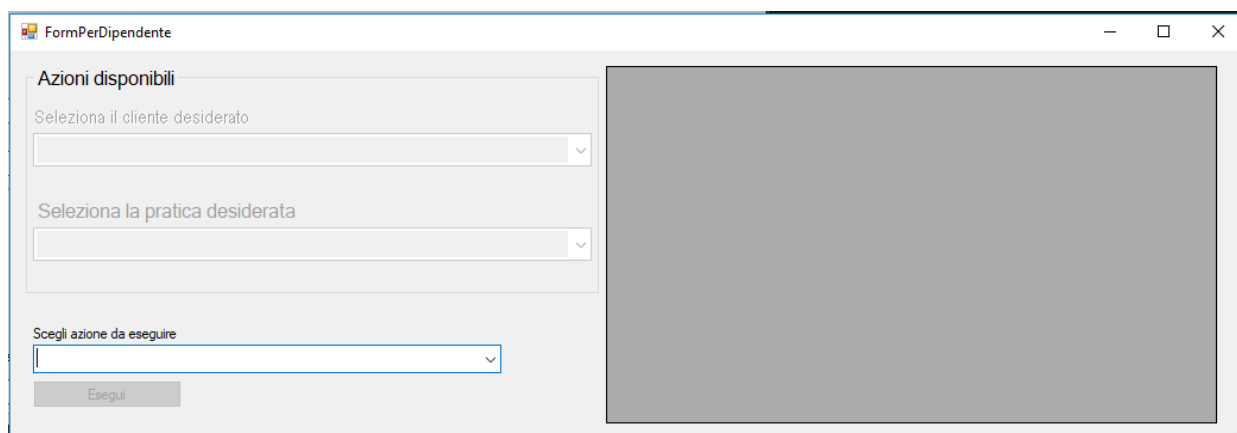
- Data pagamento:** A date input field showing "09-07-2018" with a calendar icon.
- Costo:** A numeric input field with a spinner, showing "0".
- Tipo:** A dropdown menu.
- Quantità:** A numeric input field with a spinner, showing "0".
- Sede:** A dropdown menu.
- Descrizione:** A large text area for entering details.
- Inserisci:** A button at the bottom right of the form.

Questa schermata è accessibile a entrambi i tipi di utente.

Fondamentalmente è una form per l'inserimento dei costi struttura, cercando di limitare i possibili sbagli dell'utente per evitare di creare inconsistenze nel database si è voluto implementare a livello applicativo alcune restrizioni riguardo il Tipo e la Sede, ovvero essendoci tre tipologie di costi struttura e solo quella "Materiale" ha la possibilità di poter specificare la sua quantità allora scegliendo una tipologia diversa da "Materiale" non è possibile accedere al campo Quantità.

Oltretutto il dipendente potrà solo inserire i costi struttura nella sede nella quale lavora, essendoci due sedi se il dipendente lavorasse per entrambe, le vedrebbe entrambe, se lavorasse solo per una di queste vedrebbe solo quella in cui lavora.

#### 5.1.4 - Schermata Form per dipendente:

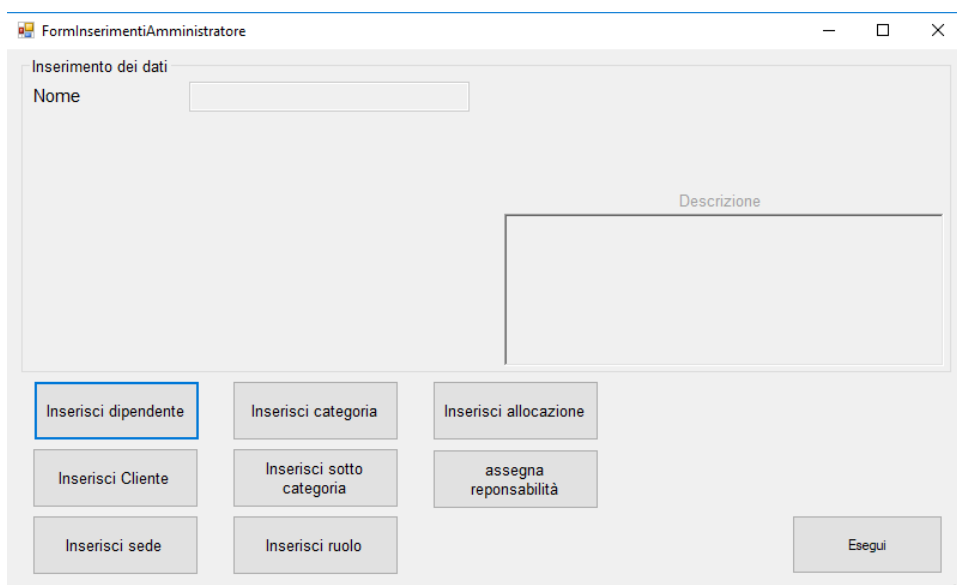


Anche questa schermata è accessibile a tutte e due i tipi di utente.

In questa schermata si possono eseguire le query per i dipendenti, selezionando il tipo di azione da eseguire, automaticamente si renderanno accessibili i campi utili per il corretto funzionamento dell'azione scelta.

Nel DataGridView a destra verranno visualizzati i risultati della query lanciata.

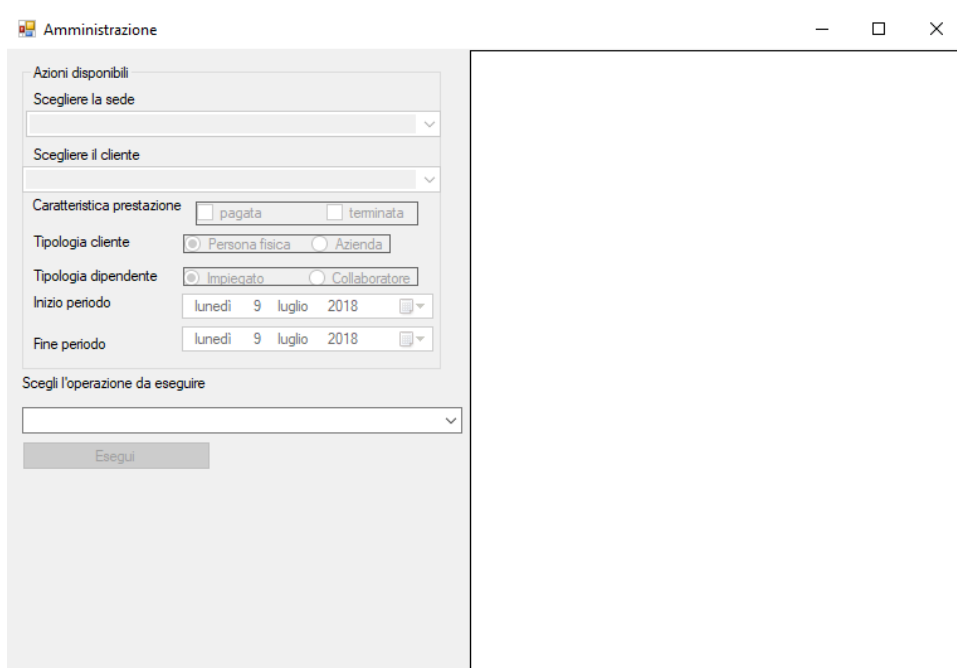
#### 5.1.5 - Schermata Amministratore Inserimenti:



Questa schermata è utilizzata solo dall'amministratore dove può effettuare tutte le operazioni di inserimento riguardanti tutte le tabelle del database.

Cliccando sul bottone interessato si visualizzeranno i campi per la popolazione di quella tabella, una volta finita la popolazione cliccando sul bottone "esegui" verrà lanciata la query di inserimento.

### 5.1.6 - Schermata di Interrogazioni per l'amministratore:



The screenshot shows a software window titled "Amministrazione". On the left is a form with the following fields:

- Azioni disponibili
- Scegliere la sede (dropdown menu)
- Scegliere il cliente (dropdown menu)
- Caratteristica prestazione: ☐ pagata, ☐ terminata
- Tipologia cliente: ☒ Persona fisica, ☐ Azienda
- Tipologia dipendente: ☒ Impiegato, ☐ Collaboratore
- Inizio periodo: lunedì 9 luglio 2018 (calendar icon)
- Fine periodo: lunedì 9 luglio 2018 (calendar icon)
- Scegli l'operazione da eseguire (dropdown menu)
- Esegui button

The right side of the window is a large, empty rectangular area, which is the DataGridView mentioned in the text.

Questa schermata sempre visibile solo all'amministratore permette la possibilità di interrogare il database con query più complesse rispetto al solo inserimento o visualizzazione.

Scegliendo l'operazione da eseguire si potrà accedere ai campi utili per il corretto funzionamento della query e i risultati ottenuti si potranno visualizzare nella DataGridView sulla destra.