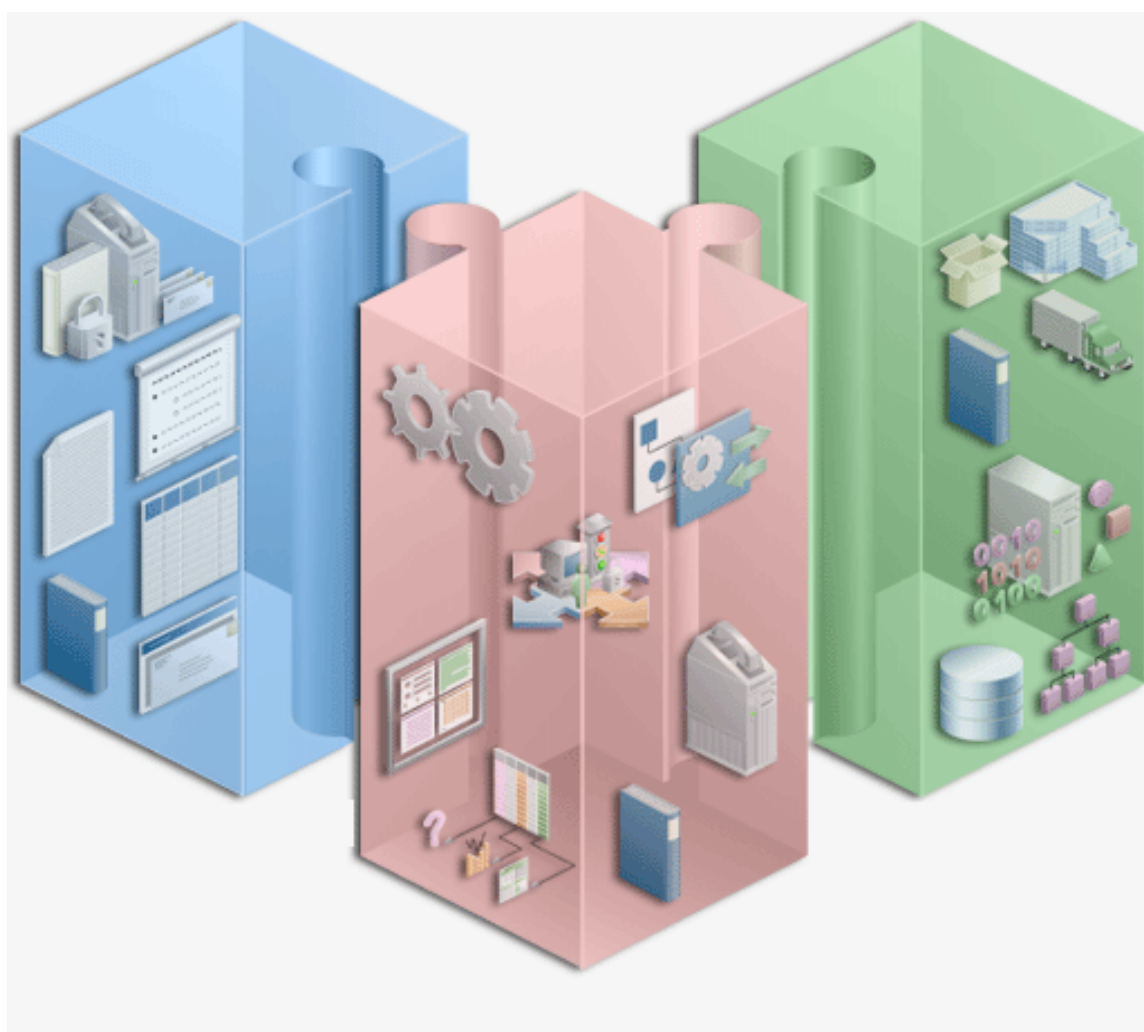

GESTIONE PROTOCOLLO INFORMATIZZATO

TESINA DI BASI DI DATI E LABORATORIO

a cura di *Gaspare Di Giovanna* e *Pier Francesco Bono*
professori *Vincenzo La Franca* e *Sabrina Mantaci*



Progettazione Concettuale

Analisi del testo e delle specifiche

In questa fase analizzeremo il testo del progetto per ricostruire le informazioni necessarie, eliminando le ambiguità delle "parole", e visualizzare in maniera più chiara quelle che sono le entità, le relazioni e le azioni.

GLOSSARIO DEI TERMINI :

TERMINI	DESCRIZIONE	SINONIMI	COLLEGAMENTI
Azienda	Struttura che necessita dell'applicativo per la gestione della sezione Protocollo	Ente	Area
Area	Entità dotata di organi di governo e gestione, di poteri di spesa e di organizzazione autonomi	AOO	Azienda, Ufficio
Ufficio	Complesso di risorse umane e strumentali cui sono affidate competenze omogenee	UOR	Area, Responsabile, Impiegato
Responsabile	Impiegato di un Ufficio che riceve in affidamento un documento o che produce documentazione da inviare	RPA	Ufficio, Documento
Documento	Lettera, email o file contenente informazioni utili all'Azienda	Lettera, Email, Fax	Responsabile, Fascicolo, Persone
Fascicolo	Insieme organico di documenti relativi ad un medesimo affare o procedimento		Responsabile, Documento
Persone	Persone che ricevono o inviano Documenti. Possono essere interne all'azienda (Impiegati) o esterne.	Impiegato, Destinatario, Mittente, Responsabile, Operatore, Dipendente, Utente	Ufficio, Responsabile

NORMALIZZAZIONE DEL TESTO :

Per semplificare il nostro lavoro, risulta utile raggruppare le specifiche in gruppi di frasi omogenee, relative cioè agli stessi concetti, ed utilizzare un linguaggio coerente e semplice.

In questo modo siamo sicuri di eliminare eventuali ambiguità e ripetizioni; otteniamo inoltre la strutturazione delle specifiche sui dati.

FRASI DI CARATTERE GENERALE

L'obiettivo è quello di realizzare un DB per un applicativo relativo alla Gestione del Protocollo Informatizzato di un'Azienda.

FRASI RELATIVE ALLA STRUTTURA DELL'AZIENDA

L'Azienda si compone di Aree.

Ciascuna Area è costituita da Uffici.

In ogni Ufficio lavorano diversi Impiegati di cui uno è il Responsabile al Protocollo.

AREE

Un'Area o AOO (Area Organizzativa Omogenea) è un'entità dotata di organi di governo e gestione, di poteri di spesa e di organizzazione autonomi ed è costituita da diversi Uffici.

Ciascuna Area ha un unico PUNTO DI REGISTRATURA, un Ufficio addetto alla registrazione dei Documenti In Arrivo destinati ad un Impiegato della stessa Area.

UFFICI

Un Ufficio o UOR (Unità Organizzativa Responsabile) è un complesso di risorse umane e strumentali cui sono affidate competenze omogenee.

In ciascun Ufficio lavorano diversi Impiegati di cui uno è il Responsabile al Protocollo. Gli Uffici appartenenti ad una stessa Area comunicano tra di loro attraverso i Documenti Interni.

FRASI RELATIVE AI DOCUMENTI

Come Documento si intende una lettera, una mail o un file contenente informazioni utili all'Azienda.

Esistono tre tipi di Documento:

- 1) *IN ARRIVO*: proveniente dall'esterno dell'azienda
- 2) *IN PARTENZA*: prodotto dall'azienda per un destinatario esterno
- 3) *INTERNO*: documento scambiato tra Uffici di una stessa Area

Ogni Documento ha un Responsabile al Protocollo. I Documenti devono essere protocollati con un numero progressivo che viene resettato ogni primo dell'anno. Ogni documento viene classificato secondo *categorie* determinate che possono essere diverse per Aree diverse.

Ogni Documento ha, nel tempo, diversi stati/etichette:

- *IN REGISTRAZIONE*: ricevuto ma non ancora registrato
- *SMISTATO*: assegnato all'Area di competenza
- *FASCICOLATO*: inserito nei giusti fascicoli

Bisogna avere la possibilità di conoscere lo stato attuale di un Documento e di sapere quali stati ha avuto in passato.

Per un Documento In Arrivo o Interno, in fase di smistamento bisogna inviare una notifica ai destinatari di ricezione della notifica. Il destinatario deve inoltre rispondere comunicando di aver correttamente fascicolato il Documento.

DOCUMENTI IN ARRIVO

Un Documento proveniente dall'esterno dell'Azienda viene *registrato* nel Punto di Registrazione dell'Area a cui fa riferimento dal Responsabile addetto.

Una volta registrato, viene poi smistato all'Ufficio di competenza ed assegnato al Responsabile dello stesso che lo visiona e lo inserisce in un Fascicolo.

DOCUMENTI IN PARTENZA

Un Documento In Partenza prodotto da un Impiegato deve essere *registrato* dal Responsabile Al Protocollo dello stesso Ufficio; quest'ultimo deve anche provvedere alla creazione e gestione del Fascicolo relativo.

DOCUMENTI INTERNI

Un Documento Interno è un Documento scambiato tra diversi Uffici di una stessa Area.

Esistono due tipi di Documento Interno:

- *INFORMATIVO*
- *GIURIDICO-PROBATORIO*

Questo Documento deve essere *registrato* dal Responsabile al Protocollo che è anche incaricato della creazione e gestione del Fascicolo relativo.

PERSONE

Persone che ricevono o inviano Documenti.

Si dividono in "persone interne all'azienda", che identifichiamo come Impiegati, e persone esterne a questa, gli Esterni.

Tutte le persone sono presenti nel database dell'azienda e identificate univocamente tramite un ID. Assumiamo che quando occorre un nuovo Esterno, presente come mittente di un Documento, questo viene inserito nel database e gli viene assegnato un nuovo ID.

IMPIEGATI

Gli Impiegati sono Persone che lavorano in un Ufficio dell'Azienda.

Ciascun Impiegato è dotato di *username* e *password* con i quali si identifica nel sistema.

Esistono quattro tipi di Impiegati:

- 1) *UTENTE*: può solo consultare una o più Aree
- 2) *OPERATORE*: può vedere, inserire e modificare i documenti della sua Area
- 3) *SUPER USER*: ha potere di Amministratore relativamente alla sua Area
- 4) *AMMINISTRATORE*: ha potere su tutte le Aree

Quando un Impiegato entra nel sistema, viene registrato l'orario e la data del login; simmetricamente vengono registrate le informazioni del logout.

Un Impiegato particolare è il Responsabile al Protocollo, sul quale è incentrata gran parte del nostro lavoro.

OPERAZIONI SUI DATI :

Accanto alle specifiche sui dati, esprimiamo le specifiche sulle operazioni da effettuare su questi dati.

Operazione 1 Trova il numero di documenti inviati da una persona.

Operazione 2 Trova il numero di documenti ricevuti da una persona.

Operazione 3 Ricerca i protocolli dei documenti per data/mittente/ambito.

Operazione 4 Ricerca i protocolli dei documenti per destinatario.

Operazione 5 Ricerca i protocolli dei documenti per responsabile.

Operazione 6 Ricerca dei documenti protocollati durante il giorno.

Operazione 7 Ricerca della cronologia di un documento.

Operazione 8 Registrazione documento.

Operazione 9 Smistamento documento.

Operazione 10 Fascicolazione documento.

Operazione 11 Registrazione logs.

SCHEMA CONCETTUALE :

Per la realizzazione del nostro schema abbiamo utilizzato una strategia ibrida.

PASSO 1:

Per la progettazione del nostro schema concettuale, siamo partiti analizzando quella che è la struttura dell'Azienda e l'abbiamo sviluppata nelle sue componenti.

L'Azienda è infatti suddivisa in Aree ma, non avendo necessità delle informazioni relative all'Azienda, non abbiamo creato un'entità per questa.

A loro volta le Aree sono costituite da Uffici, abbiamo quindi creato le entità relative e le abbiamo legate tramite la relazione Composizione.

PASSO 2:

Una delle entità per cui abbiamo incontrato una certa difficoltà è Persone, composta da una generalizzazione importante per l'azienda, Impiegati, e un'altra che serve per identificare persone non interne all'azienda, Esterni.

Impiegati si compone di 4 diverse generalizzazioni che identificano il tipo di impiegato: Utenti, Operatori, Super Users e Amministratori.

Abbiamo assunto che le entità Operatori, Super Users e Amministratori possono a loro volta possedere la caratteristica definita dall'entità Responsabile.

Abbiamo legato Impiegati a Uffici tramite la relazione Afferenza.

PASSO 3:

Un'altra entità molto importante è quella relativa a Documenti, che si suddivide in "In arrivo", "In partenza" e "Interni". Questi ultimi si distinguono a loro volta in Giuridici ed Informativi.

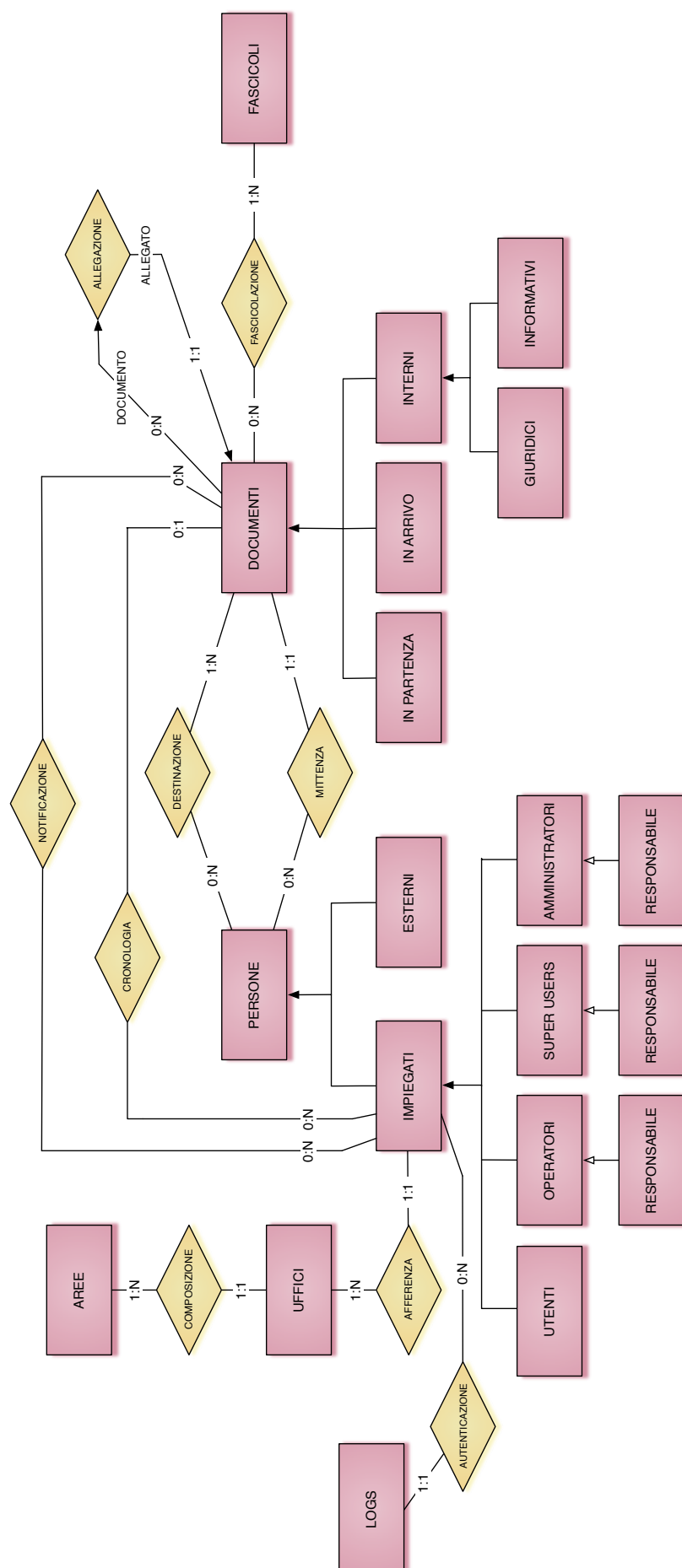
L'entità Fascicoli è collegata con l'entità Documenti per mezzo della relazione Fascicolazione.

Documenti è inoltre collegata a se stessa mediante la relazione Allegazione.

PASSO 4:

Abbiamo collegato le entità Documenti e Persone per mezzo delle relazioni Mittenza e Destinazione.

Abbiamo legato le entità Impiegati con Documenti con la relazione Cronologia e Notificazione e l'entità Impiegati con Logs (che gestisce i dati relativi alle informazioni di login e logout) attraverso la relazione Autenticazione.



DIZIONARIO DEI DATI :

ENTITÀ	DESCRIZIONE	ATTRIBUTI	IDENTIFICATORE
Aree	Informazioni sull'area.	id area, nome	id area
Uffici	Informazioni sull'ufficio.	id ufficio, nome, pt registratura	id ufficio
Persone	Informazioni sulle persone che hanno a che fare con l'azienda.	id persona, nome, cognome, ragione sociale	id persona
Impiegati	Specializzazione dell'entità Persone. Identifica gli impiegati afferenti agli uffici.	username, password	Persone.id persona
Documenti	Informazioni sui documenti.	id documento, oggetto, data ricez./trasm., mezzo ricez./trasm., protocollo, anno, categoria, stato	id documento
Fascicoli	Informazioni sui fascicoli.	id fascicolo, descrizione	id fascicolo
Logs	Informazioni sul login e logout degli impiegati dal sistema	data login, data logout	Impiegato.username, data login

RELAZIONE	DESCRIZIONE	ENTITÀ COINVOLTE	ATTRIBUTI
Composizione	Associa ogni area ai suoi uffici.	Aree (1:N), Uffici (1:1)	
Afferenza	Associa ogni ufficio con gli impiegati che ci lavorano.	Uffici (1:N), Impiegati (1:1)	

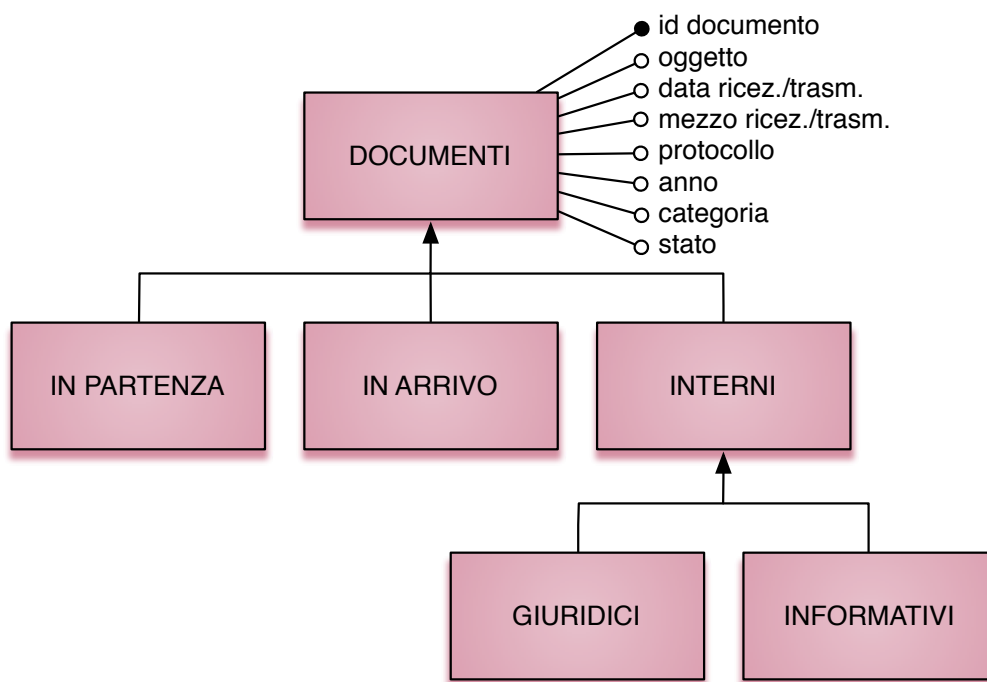
RELAZIONE	DESCRIZIONE	ENTITÀ COINVOLTE	ATTRIBUTI
Mittenza	Relazione che identifica il mittente di ciascun documento.	Documenti (1:1), Persone (0:N)	
Destinazione	Relazione che identifica i destinatari di un documento.	Documenti (1:N), Persone (0:N)	
Cronologia	Relazione tra Documenti e Impiegati che definisce lo storico di un documento e chi ha provocato il cambio di stato.	Documenti (0:1), Impiegati (0:N)	data registrazione, data smistamento, data fascicolazione
Allegazione	Relaziona un documento ai suoi allegati (che sono sempre documenti).	Documenti (1:1) , [referente] Documenti (0:N), [riferiti]	
Fascicolazione	Associa ad ogni fascicolo i suoi documenti.	Documenti (0:N), Fascicoli (1:N)	
Autenticazione	Relazione tra Impiegati e Logs che memorizza i tempi di login e logout degli Impiegati dal sistema	Logs (1:1), Impiegati (0:N)	
Notificazione	Relazione tra Documenti e Impiegati che segnala agli impiegati i documenti di cui devono prendere visione o l'eventuale messaggio di avvenuta notifica.	Documenti (0:N), Impiegati (0:N)	notifica

ELIMINAZIONE DELLE GERARCHIE :

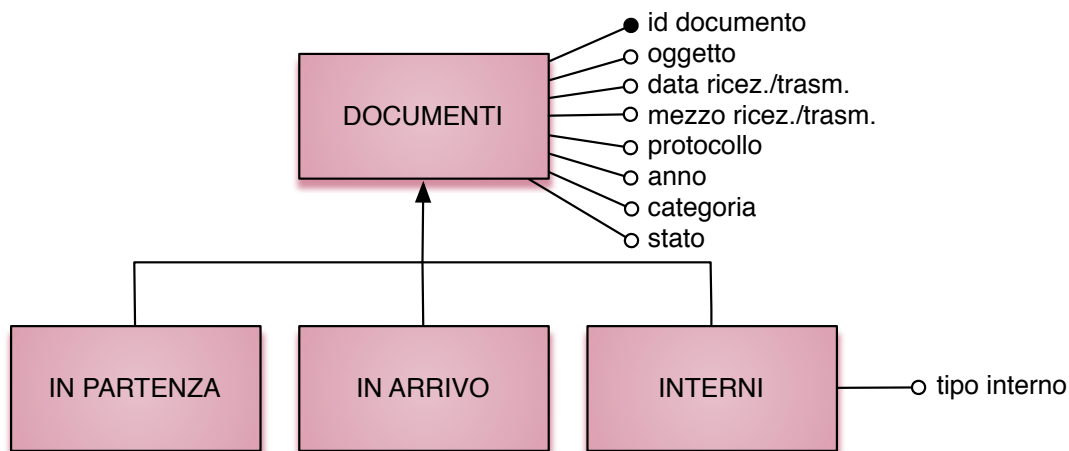
Nel nostro schema relazionale abbiamo 5 generalizzazioni.

L'entità DOCUMENTI è divisa in "IN PARTENZA", "IN ARRIVO" e "INTERNI" (1a generalizzazione).

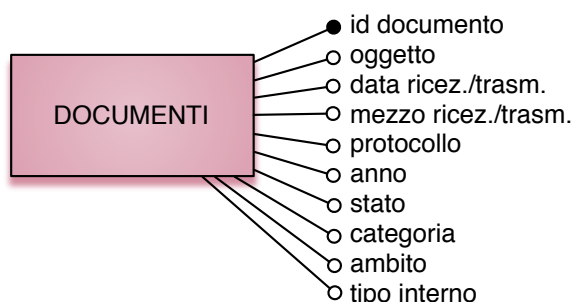
A sua volta "INTERNI" è specializzata in "GIURIDICI" e "INFORMATIVI" (2a generalizzazione).



Dato che i documenti *interni* possono essere soltanto di due tipi, GIURIDICO/PROBATORI e INFORMATIVI, e che questi non contengono particolari attributi, possiamo accorparli inserendo in INTERNI un attributo che permette di distinguerli.



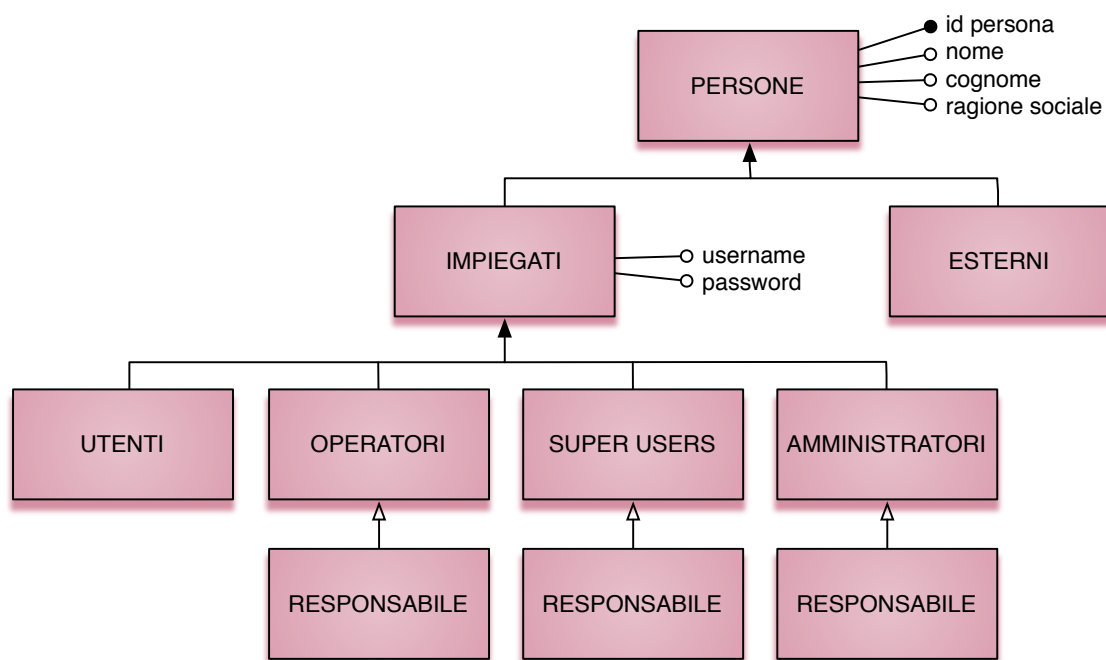
Per lo stesso motivo possiamo accorpare a DOCUMENTI le tre tipologie in cui questi si dividono. Dobbiamo però portarci dietro l'attributo "tipo interno" di INTERNI (questo sarà utilizzato quando avremo il corretto tipo di documento).



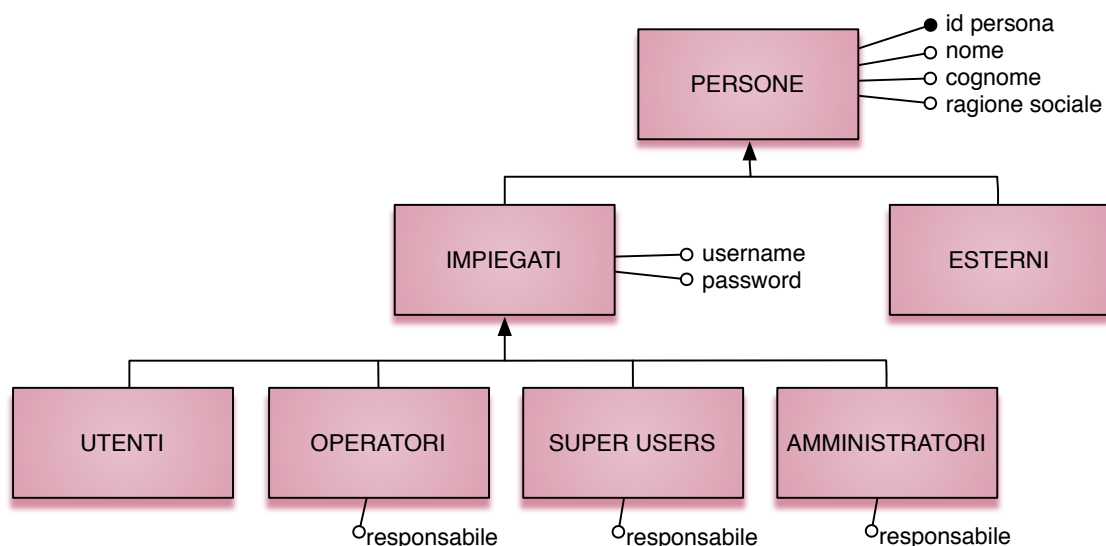
L'entità PERSONE è divisa in "IMPIEGATI" e "ESTERNI" (3a generalizzazione).

IMPIEGATI a sua volta è specializzato in "UTENTI", "OPERATORI", "SUPER USERS" e "AMMINISTRATORI" (4a generalizzazione).

Le tre entità "OPERATORI", "SUPER USERS" e "AMMINISTRATORI" presentano la generalizzazione comune "RESPONSABILE" (5a generalizzazione).

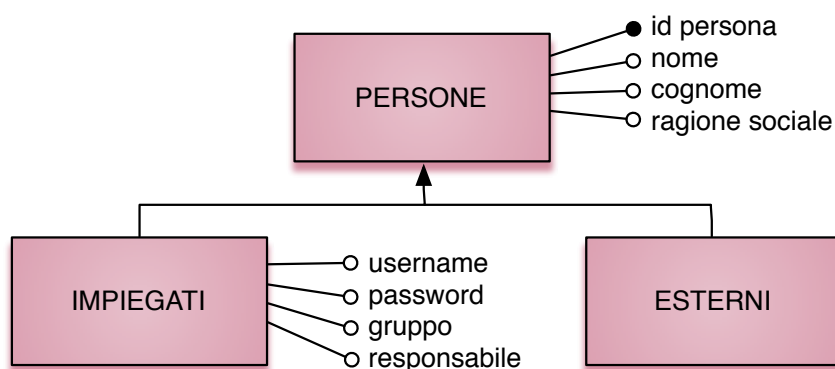


possiamo fare un semplice accorpamento di RESPONSABILE nei tre tipi di impiegati che possono esserlo, inserendo in questi una flag che può assumere due valori (0,1).



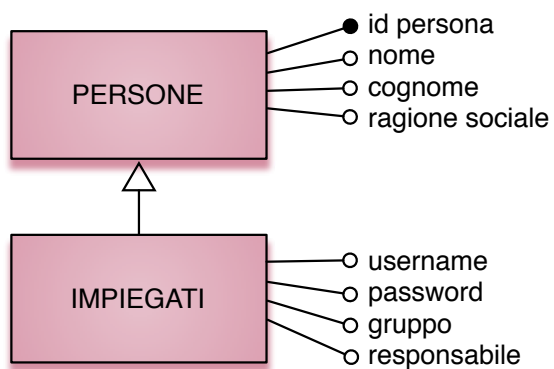
Si può ridurre ulteriormente accorpando i 4 tipi di impiegato presenti ad IMPIEGATI, inserendo un attributo opportuno, che chiamiamo *gruppo*, che imporrà potrà prendere valore nell'insieme ('UTENTE', 'OPERATORE', 'SUPER USER', 'AMMINISTRATORE').

Controlleremo che la flag *responsabile* potrà avere valore '1' soltanto se ci troveremo in uno dei gruppi opportuni.

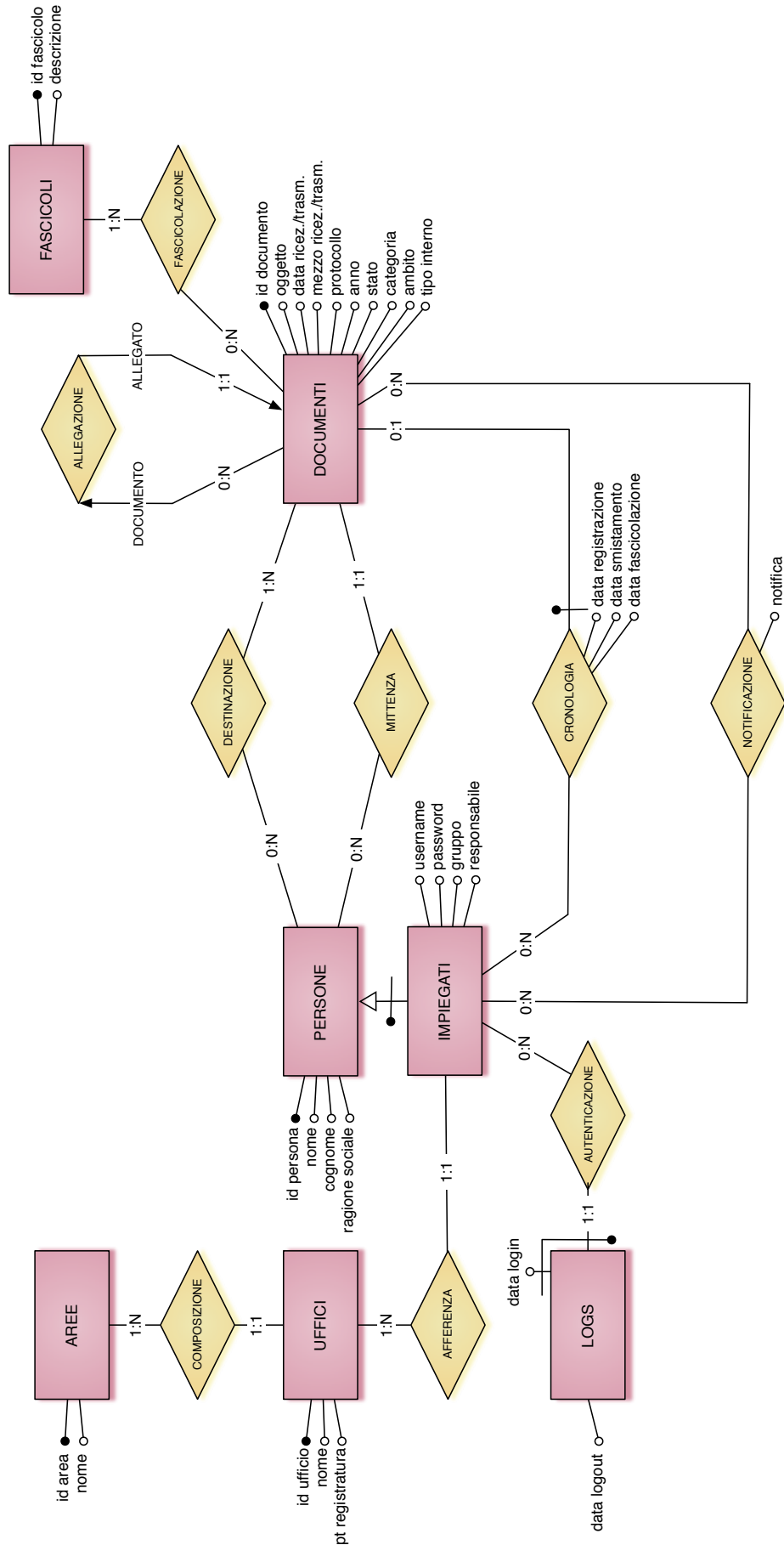


Poiché l'entità Esterni non presenta attributi in più rispetto all'entità Persone, possiamo accorpare l'entità figlio al genitore. L'entità Impiegati non la accorpriamo a Persone perché (se così facessimo) allocheremmo spazio in memoria per dei campi (username, password, gruppo, responsabile) inutilizzato per gli Esterni. Questo ci facilita la ricerca di informazioni riguardanti gli impiegati.

Risulta inoltre più efficiente relazionare le entità Impiegati con Uffici, Cronologia, Logs e Documenti.



Adesso IMPIEGATI diventa una generalizzazione di PERSONE e la sua chiave sarà la chiave di quest'ultima.



PROGETTAZIONE LOGICA :

Lo scopo di questa sezione è quello di semplificare la traduzione fatta durante la progettazione concettuale e ottimizzare il progetto.

Ristrutturazione dello schema concettuale

Per effettuare questa operazione di ristrutturazione dobbiamo specificare il volume dei dati e le caratteristiche delle operazioni da effettuare in maniera da poter meglio valutare l'efficienza dello schema.

TAVOLA DEI VOLUMI

CONCETTO	TIPO	VOLUME
Aree	E	6
Uffici	E	18
Impiegati	E	360
Persone	E	10000
Documenti	E	400000
Allegazione	R	100000
Cronologia	R	400000
Fascicoli	E	40000
Destinazione	R	800000
Notificazione	R	5000

- Ci sono circa 6 Aree nell'Azienda.
- Ogni Area ha in media 3 Uffici.
- In ogni Ufficio lavorano in media 20 Impiegati.
- Supponiamo che abbiamo circa 10000 persone (impiegati compresi) presenti nel database dell'azienda.
- Abbiamo un volume di 50000 Documenti l'anno per Area, supponiamo che di questi il 10% siano interni e il restante 90% sia bilanciato tra "In Arrivo" e "In Partenza".
- Ipotizziamo che in media un documento su tre abbia un allegato (anch'esso un documento). Abbiamo quindi circa 400000 documenti di cui

100000 sono allegati.

- Supponiamo che in media in ogni fascicolo abbia 50 documenti, ma esistono fascicoli che hanno gli stessi documenti. Ipotizzando che uno stesso documento possa essere presente in media in 5 fascicoli. Abbiamo quindi un volume di circa 40000 fascicoli.
- Supponiamo che ogni documento abbia in media 2 destinatari, abbiamo quindi un volume di circa 800000 destinatari.
- Supponiamo che Notificazione possa avere un volume di dati di 5000 notifiche, dato che essenzialmente è una tabella temporanea.

TAVOLA DELLE OPERAZIONI

OPERAZIONE	TIPO	FREQUENZA
Operazione 1	I	50 al giorno
Operazione 2	I	50 al giorno
Operazione 3	I	500 al giorno
Operazione 4	I	100 al giorno
Operazione 5	I	100 al giorno
Operazione 6	B	1 al giorno
Operazione 7	I	30 al giorno
Operazione 8	I	1100 al giorno
Operazione 9	I	1100 al giorno
Operazione 10	I	1100 al giorno
Operazione 11	I	1000 al giorno

Op. 1 Trova il numero di documenti inviati da una persona.

Op. 2 Trova il numero di documenti ricevuti da una persona.

Op. 3 Ricerca i protocolli dei documenti per data/mittente/ambito.

Op. 4 Ricerca i protocolli dei documenti per destinatario.

Op. 5 Ricerca i protocolli dei documenti per responsabile.

Op. 6 Ricerca dei documenti protocollati durante il giorno.

Op. 7 Ricerca della cronologia di un documento.

Op. 8 Registrazione documento.

Op. 9 Smistamento documento.

Op. 10 Fascicolazione documento.

Op. 11 Registrazione logs.

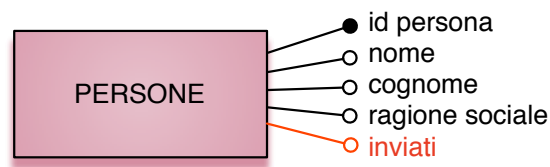
Analisi delle Prestazioni:

In questa fase analizzeremo le prestazioni del nostro database e valuteremo l'eventuale aggiunta di attributi ridondanti o la rimozione di questi.

Abbiamo considerato il peso dell'operazione *lettura* (L) come unitario; il costo dell'operazione *scrittura* (S) è doppio rispetto alla prima.

OPERAZIONE 1 :Trova il numero di documenti inviati da una persona.

Possiamo valutare di inserire l'attributo "inviati" nell'entità persone per migliorare le prestazioni di questa operazione.



OPERAZIONE 1 (SENZA RIDONDANZA)

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Documenti	E	400000	L

OPERAZIONE 1 (CON RIDONDANZA)

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Persone	E	1	L

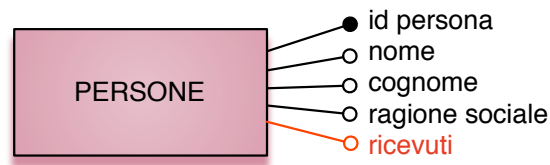
L'operazione 1 viene effettuata mediamente 50 volte al giorno e in assenza di ridondanza avremmo $50 \times 400000 = 20$ milioni di accessi in lettura, in presenza di ridondanza avremmo un totale di 50 accessi in lettura.

In fase di invio di un documento si dovrebbe aggiornare il campo *inviati* di Documenti. Avremo circa 1100 aggiornamenti al giorno di tale campo, che comporterebbe un totale di 1100 accessi in scrittura equivalenti a 2200 accessi in lettura.

Decidiamo quindi che sia più conveniente aggiungere la ridondanza *inviati*.

OPERAZIONE 2 : Trova il numero di documenti ricevuti da una persona.

Possiamo valutare di inserire l'attributo *ricevuti* nell'entità persone per migliorare le prestazioni di questa operazione.



OPERAZIONE 2 (SENZA RIDONDANZA)

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Destinatari	R	800000	L

OPERAZIONE 2 (CON RIDONDANZA)

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Persone	E	1	L

L'operazione 2 viene effettuata mediamente 50 volte al giorno e in assenza di ridondanza avremmo $50 \times 800000 = 40$ milioni di accessi in lettura, in presenza di ridondanza avremmo un totale di 50 accessi in lettura.

Avremo circa 1100 documenti al giorno con una media di 2 destinatari. Dovremo effettuare in media 2200 accessi in scrittura al giorno (equivalenti a 4400 accessi in lettura) per aggiornare la ridondanza *ricevuti*.

Decidiamo quindi che sia più conveniente aggiungere la ridondanza *ricevuti*.

OPERAZIONE 3 : Ricerca i protocolli dei documenti per data/mittente/ambito.

OPERAZIONE 3

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Documenti	L	400000	L

L'operazione è effettuata in media 500 volte al giorno, abbiamo quindi $400000 \times 500 = 200$ milioni di accessi in lettura. Possiamo prevedere l'utilizzo di indicizzazioni per questi campi in maniera da ottimizzare questo tipo di ricerche.

OPERAZIONE 4 : Ricerca i protocolli dei documenti per destinatario.

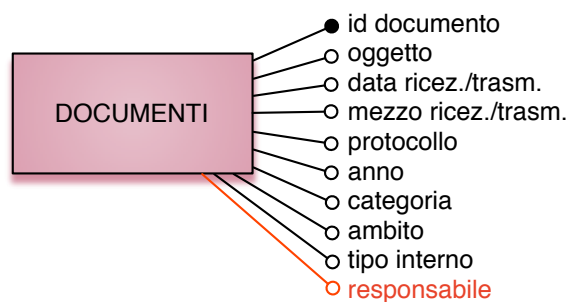
OPERAZIONE 4

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Destinatari	R	800000	L
Documenti	E	400000	L

L'operazione è effettuata in media 100 volte al giorno, abbiamo quindi $(800000 + 400000) \times 100 = 120$ milioni di accessi in lettura. Prevediamo l'utilizzo di un indice per *destinatario* in Destinatari per ottimizzare questo tipo di ricerche.

OPERAZIONE 5 : Ricerca i protocolli dei documenti per responsabile.

Per questa operazione possiamo valutare l'inserimento della ridondanza *responsabile* in documenti.



OPERAZIONE 5 (SENZA RIDONDANZA)

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Cronologia	R	400000	L
Documenti	E	400000	L

La ricerca del responsabile deve essere fatta all'interno della relazione Cronologia perché abbiamo ipotizzato che il "responsabile del documento" è colui che ne effettua l'operazione di registrazione.

OPERAZIONE 5 (CON RIDONDANZA)

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Documenti	E	400000	L

L'operazione è effettuata in media 100 volte al giorno, abbiamo quindi $(400000 + 400000) \times 100 = 80$ milioni di accessi in lettura senza ridondanza; in presenza di ridondanza otterremmo un totale di 40 milioni di accessi in lettura.

Se inseriamo la ridondanza, in fase di protocollo dovremmo inserire l'id del registratore in Documenti che comporterebbe in media 1100 accessi in scrittura al giorno, equivalenti a 2200 accessi in lettura. Essendo l'operazione di Registratura comunque performata, l'aggiornamento del campo risulta trascurabile così come la memoria occupata.

Decidiamo quindi che risulta più conveniente utilizzare la ridondanza *responsabile* e, per migliorare l'efficienza del nostro DB, prevediamo l'utilizzo di un indice per questo.

OPERAZIONE 6 : Ricerca dei documenti protocollati durante il giorno.

Avendo inserito *data_registrazione* all'interno di Cronologia, dobbiamo prima effettuare la ricerca all'interno di questa e successivamente ottenere le informazioni dei protocolli da Documenti.

OPERAZIONE 6

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Cronologia	R	400000	L
Documenti	E	400000	L

Essendo questa operazione effettuata una volta al giorno, abbiamo in totale $(400000 + 400000) \times 1 = 800000$ accessi in lettura.

Potremmo dimezzare gli accessi in lettura inserendo gli attributi di Cronologia all'interno di Documenti ma abbiamo optato per la soluzione di mantenere i due concetti separati anche se tra essi esiste una relazione 1:1.

Prevediamo l'indicizzazione di *data_registrazione* per migliorare le prestazioni per questa operazione.

OPERAZIONE 7 : Ricerca della cronologia di un documento.

OPERAZIONE 7

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Cronologia	R	1	L

Quest'operazione è eseguita in media 30 volte al giorno, abbiamo in totale $1 \times 30 = 30$ accessi in lettura.

OPERAZIONE 8 : Registrazione documento.

OPERAZIONE 8

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Documenti	E	1	S
Cronologia	R	1	S

Quest'operazione è eseguita in media 1100 volte al giorno, abbiamo in totale $(1 + 1) \times 1100 = 2200$ accessi in scrittura, che equivalgono a 4400 accessi in lettura.

OPERAZIONE 9 : Smistamento documento.

OPERAZIONE 9

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Destinatari	E	2	L
Notificazione	R	2	S

Per smistare un documento, si deve accedere a Destinatari da cui prendere i destinatari del documento.

Per ogni documento abbiamo in media 2 destinatari, dobbiamo quindi effettuare 2 operazioni di scrittura in Notificazione.

L'operazione è seguita in media 1100 volte al giorno, abbiamo in totale $(2 + 4) \times 1100 = 6600$ accessi in lettura (il 4 indica i due accessi in scrittura).

OPERAZIONE 10 : Fascicolazione documento.**OPERAZIONE 10**

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Cronologia	R	1	S
Fascicolo	E	1	S
Fascicolazione	R	5	S

In fase di fascicolazione viene aggiornata la cronologia del documento e prevediamo l'eventuale creazione di un nuovo fascicolo.

Dato che in media un documento può essere presente in 5 fascicoli, dobbiamo collegare il documento a questi.

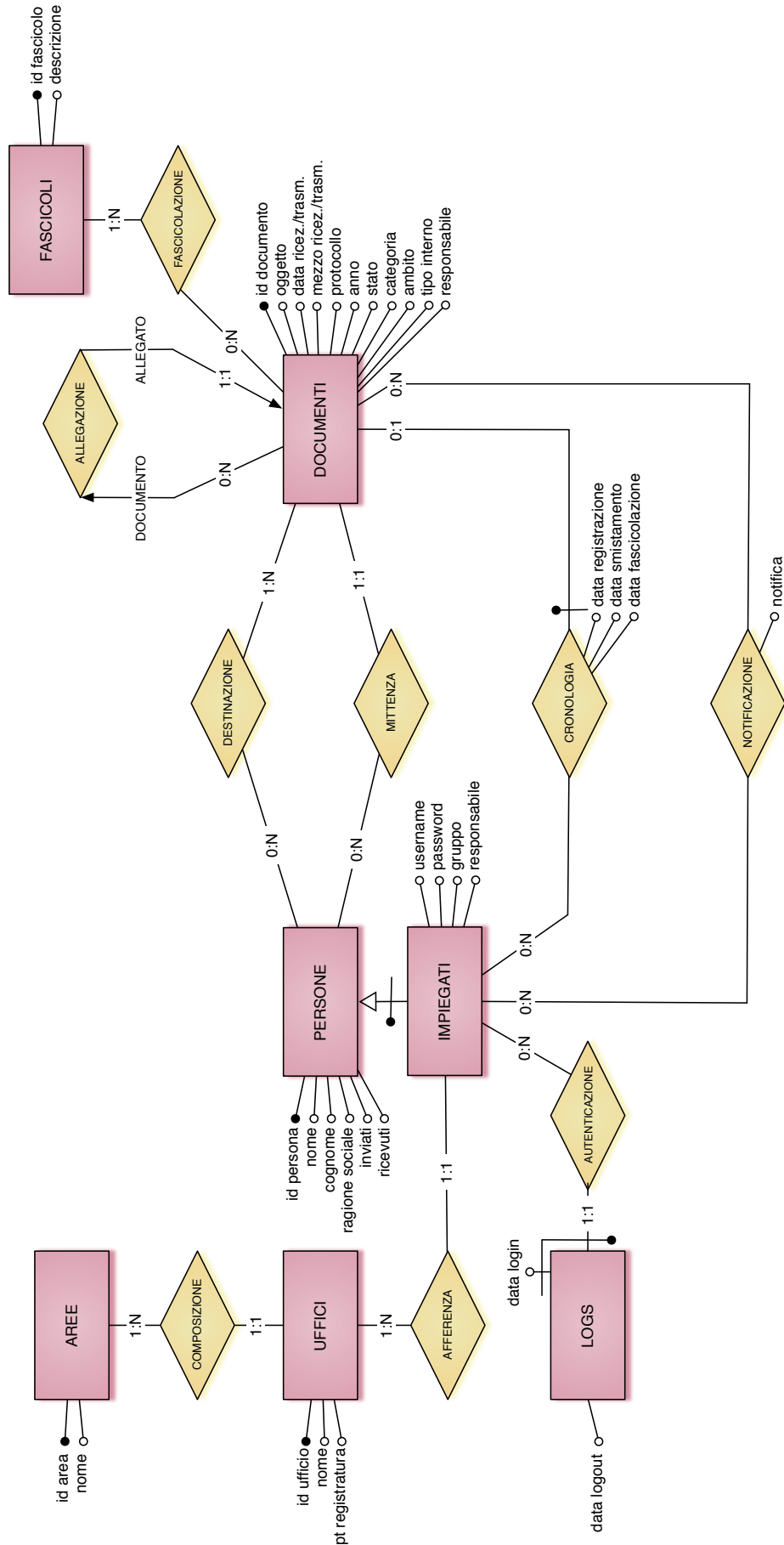
In totale l'operazione costa $(1 + 1 + 5) \times 1100 = 7700$ accessi in scrittura che equivalgono a 15400 accessi in lettura.

OPERAZIONE 11 : Registrazione Logs.**OPERAZIONE 12**

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Logs	E	2	S

Login e *logout* avvengono in tempi diversi ma, considerato il fatto che al primo deve obbligatoriamente seguire il secondo, ne abbiamo tenuto conto nei nostri calcoli.

L'operazione è eseguita in media 1000 volte al giorno, abbiamo quindi un totale di $2 \times 1000 = 2000$ accessi in scrittura equivalenti a 4000 accessi in lettura.



Scelta degli identificatori principali

All'entità Fascicoli abbiamo aggiunto la proprietà *id_fascicolo* che sarà la chiave primaria.

Fascicolazione ha come chiave la coppia *fascicolo* e *documento*.

All'entità Area abbiamo aggiunto la proprietà *id_area* che sarà la chiave primaria.

Le entità Persone e Impiegati prevedono che la chiave possa essere costituita dalle 3 proprietà *nome*, *cognome* e *rag_sociale*, scelta non conveniente sia in termini di prestazioni del database. Aggiungiamo quindi una proprietà *id_persona* che sarà la chiave primaria per le nostre entità. Dato che abbiamo scelto di mantenere due tabelle separate per Impiegati e Persone, avremo l'identificatore *id_impiegato* in Impiegati che sarà un riferimento alla chiave dell'altra tabella.

Un ragionamento analogo è stato fatto per l'entità Documenti, ove le nostre istanze sarebbero potute essere individuate per mezzo delle proprietà *mittente*, *oggetto* e *data_ricezione*. Per far sì che la chiave fosse costituita da un unico attributo abbiamo dotato la nostra entità di *id_documento*. Così facendo anche l'identificatore principale dell'entità Cronologia è *id_documento*.

Visto che un documento può avere più destinatari, il concetto *Destinatari* come chiave primaria avrà la coppia *documento* e *destinatario*.

La tabella Allegati avrà come identificatore la coppia *documento* e *allegato*, in quanto un documento può avere più di un allegato e quindi il solo attributo *documento* non sarebbe stato sufficiente ad individuare una tupla di questa relazione.

L'entità Logs ha come chiave primaria le proprietà di *username* e *login_time*, anche qui non è sufficiente individuare una tupla con il solo attributo *username*, in quanto un impiegato effettua diverse volte il login.

In Notifiche la chiave è costituita da *documento*, *mitt_notifica* e *dest_notifica*.

Traduzione verso il Modello Relazionale.

Fascicoli (id_fascicolo, descrizione)

Fascicolati (fascicolo, documento)

Aree (id_area, nome)

Uffici (id_ufficio, nome, area, pt_registratura)

Persone (id_persona, nome, cognome, rag_sociale, inviati, ricevuti)

Impiegati (id_impiegato, ufficio, gruppo, responsabile, username, password)

Documenti (id_documento, mittente, oggetto, data_ricezione, protocollo, anno, stato, mezzo_ric_tr, categoria, ambito, tipo_interno, responsabile)

Destinatari (documento, destinatario)

Allegati (documento, allegato)

Cronologia (documento, data_registrazione, registratore, data_smistamento, smistatore, data_fascicolazione, fascicolatore)

Logs (username, login_time, logout_time)

Notifiche (documento, dest_notifica, mitt_notifica, notifica)

Vincoli di Integrità Referenziale

In Uffici *area* deve essere un *id_area* valido.

Per quanto riguarda Impiegati, *id_impiegato* deve essere un *id_persona* valido e *ufficio* deve essere un *id_ufficio* valido.

In Documenti *mittente* deve essere un *id_persona* valido e *responsabile* deve essere un *id_impiegato* valido.

Nella relazione Destinatari, *documento* deve essere un *id_documento* valido e *destinatario* deve essere un *id_persona* valido.

Nella relazione Allegati, *documento* e *allegato* devono essere degli *id_documento* validi.

In Cronologia *documento* deve essere un *id_documento* valido mentre *registratore*, *smistatore* e *fascicolatore* devono essere degli *id_impiegato* validi.

In Logs *username* deve essere un *username* di Impiegati valido.

In Fascicolati *fascicolo* deve essere un *id_fascicolo* valido e *documento* deve essere un *id_documento* valido.

In Notifiche *documento* deve essere un *id_documento* valido mentre *dest_notifica* e *mitt_notifica* devono essere degli *id_impiegato* validi.

Vincoli di Tabella

In Fascicoli l'attributo *descrizione* deve assumere valore unico.

In Aree l'attributo *nome* deve assumere valore unico.

In Uffici l'attributo *nome* deve assumere valore unico.

In Persone la terna *nome*, *cognome*, *rag_sociale* deve essere unica.

In Impiegati l'attributo *username* deve essere unico e il valore dell'attributo *gruppo* deve essere uno tra ('utente', 'operatore', 'super user', 'amministratore'). Il valore dell'attributo *responsabile* può assumere valore '0' o '1'. Se l'attributo *gruppo* ha valore *utente* allora l'attributo *responsabile* deve avere valore '0'.

In Documenti la coppia *anno* e *protocollo* deve essere unica e il valore dell'attributo *ambito* deve essere uno tra ('in arrivo', 'in partenza', 'interno'). Se *ambito* assume come valore 'interno' allora *tipo_interno* deve assumere valore 'giuridico' o 'informativo', in caso contrario deve essere NULL.

In Cronologia il valore dell'attributo *data_registrazione* deve essere minore dell'attributo *data_smistamento* che deve essere minore di *data_fascicolazione*.

In Logs l'attributo *login_time* deve assumere un valore minore di *logout_time*.

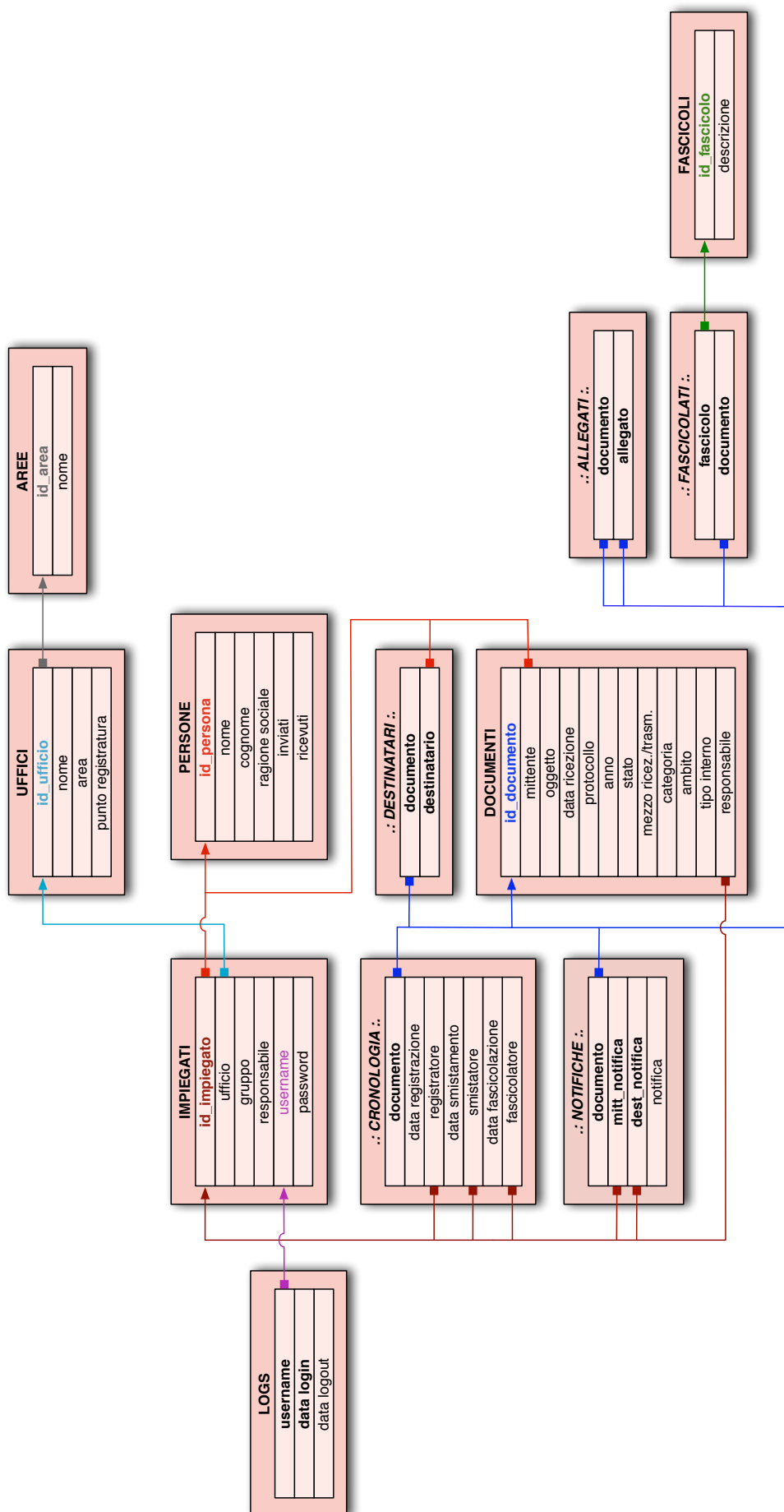
In Notifiche l'attributo *notifica* deve essere uno tra ('in notifica', 'notificato', 'risposta alla notifica').

NORMALIZZAZIONE

Il nostro database rispetta la prima forma normale in quanto non presenta gruppi di attributi che si ripetono (ossia ciascun attributo è definito su un dominio con valori atomici) ed esiste una chiave primaria per ciascuna tabella.

La nostra base di dati è in seconda forma normale poiché è in prima forma normale e, per ogni tabella, tutti i campi non chiave dipendono funzionalmente dall'intera chiave primaria e non da una parte di essa.

La nostra base di dati è in terza forma normale tranne in Documenti dove i campi *protocollo* e *anno*, dopo l'operazione di registrazione, costituiscono chiave per la nostra relazione; abbiamo tuttavia scelto di mantenerla in questo modo perché, rispettando comunque la forma normale di Boyce-Codd, risulta conveniente per il nostro sistema.



Creazione DataBase

CREAZIONE TABELLE

In questa prima parte creeremo le tabelle sulle quali dobbiamo lavorare cercando di seguire l'ordine top-down della struttura logica.

```
CREATE TABLE AREE
(
    id_area CHAR(3) PRIMARY KEY,
    nome    VARCHAR2(30) UNIQUE
)
```

```
CREATE TABLE UFFICI
(
    id_ufficio CHAR(3) PRIMARY KEY,
    nome      VARCHAR2(20) UNIQUE NOT NULL,
    area REFERENCES aree(id_area) ON UPDATE CASCADE NOT NULL,
    pt_registratura CHAR(1) DEFAULT '0' NOT NULL,
    CHECK(pt_registratura IN ('0', '1'))
)
```

```
CREATE TABLE PERSONE
(
    id_persona CHAR(10) PRIMARY KEY,
    nome       VARCHAR2(20) NOT NULL,
    cognome    VARCHAR2(20) NOT NULL,
    rag_sociale VARCHAR(20) DEFAULT NULL,
    inviati    NUMBER(6) DEFAULT 0,
    ricevuti   NUMBER(6) DEFAULT 0,
    UNIQUE(nome, cognome, rag_sociale)
)
```

```
CREATE TABLE IMPIEGATI
(
    id_impiegato REFERENCES persone(id_persona) PRIMARY KEY ON DELETE CASCADE,
    ufficio REFERENCES uffici(id_ufficio) ON DELETE SET NULL ON UPDATE CASCADE,
    gruppo      VARCHAR2(15) NOT NULL,
    responsabile CHAR(1) DEFAULT '0' NOT NULL,
    username    VARCHAR2(20) UNIQUE NOT NULL,
    password    VARCHAR2(20) NOT NULL,
    CHECK(gruppo IN ('utente', 'operatore', 'super user', 'amministratore')),
    CHECK(responsabile IN ('0', '1')),
    CHECK(responsabile = '1' AND gruppo <> 'utente')
)
```

```
CREATE TABLE DOCUMENTI
(
    id_documento CHAR(12) PRIMARY KEY,
    mittente REFERENCES persone(id_persona) ON DELETE SET NULL ON UPDATE CASCADE,
    oggetto      VARCHAR2(20)
    ,
    data_ricezione DATE
    ,
    protocollo   NUMBER(8)
    ,
    anno         NUMBER(4)
    ,
    stato        VARCHAR2(20) DEFAULT 'in registrazione' NOT NULL,
    mezzo_ric_tr VARCHAR2(20) NOT NULL
    ,
    categoria    VARCHAR2(20)
    ,
    ambito       VARCHAR2(20) NOT NULL
    ,
    tipo_interno VARCHAR2(20)
    ,
    responsabile REFERENCES impiegati(id_impiegato)
    ON DELETE SET NULL ON UPDATE CASCADE
    ,
    UNIQUE(protocollo, anno)
    ,
    CHECK(stato      IN ('in registrazione', 'registrato', 'fascicolato')),
    CHECK(ambito     IN ('in arrivo', 'in partenza', 'interno'))
    ,
    CHECK(protocollo = NULL AND anno = NULL AND stato = 'in registrazione')
    ,
    CHECK((tipo_interno IN ('informativo', 'giuridico') AND ambito = 'interno') OR
    (ambito <> 'interno' AND tipo_interno = NULL))
    )
```

```
CREATE TABLE FASCICOLI
(
    id_fascicolo CHAR(8) PRIMARY KEY,
    descrizione   VARCHAR2(20) UNIQUE NOT NULL
    )
```

```
CREATE TABLE FASCICOLATI
(
    fascicolo REFERENCES fascicoli(id_fascicolo)
    ON DELETE CASCADE ON UPDATE CASCADE,
    documento REFERENCES documenti(id_documento)
    ON DELETE CASCADE ON UPDATE CASCADE,
    PRIMARY KEY(fascicolo, documento)
    )
```

```
CREATE TABLE DESTINATARI
(
    documento REFERENCES documenti(id_documento)
    ON DELETE CASCADE ON UPDATE CASCADE,
    destinatario REFERENCES persone(id_persona)
    ON DELETE CASCADE ON UPDATE CASCADE,
    PRIMARY KEY(documento, destinatario)
    )
```

```
CREATE TABLE ALLEGATI
(
    documento REFERENCES documenti(id_documento)
    ON DELETE CASCADE ON UPDATE CASCADE,
    allegato   REFERENCES documenti(id_documento)
    ON DELETE CASCADE ON UPDATE CASCADE,
    PRIMARY KEY(documento, allegato)
    )
```

```
CREATE TABLE CRONOLOGIA
(
    documento REFERENCES documenti(id_documento)
        ON DELETE CASCADE ON UPDATE CASCADE PRIMARY KEY ,
    data_registrazione DATE NOT NULL,
    registratore REFERENCES impiegati(id_impiegato)
        ON DELETE SET NULL ON UPDATE CASCADE ,
    data_smistamento DATE,
    smistatore REFERENCES impiegati(id_impiegato)
        ON DELETE SET NULL ON UPDATE CASCADE ,
    data_fascicolazione DATE,
    fascicolatore REFERENCES impiegati(id_impiegato)
        ON DELETE SET NULL ON UPDATE CASCADE ,
    CHECK( ( ( data_fascicolazione IS NULL
        OR
        data_fascicolazione >= data_smistamento )
        AND ( data_smistamento IS NOT NULL ) )
        AND ( data_smistamento IS NULL
        OR
        data_smistamento >= data_registrazione )
    )
)
```

```
CREATE TABLE LOGS
(
    username REFERENCES impiegati(username) ON DELETE CASCADE ON UPDATE CASCADE,
    login_time TIMESTAMP ,
    logout_time TIMESTAMP ,
    PRIMARY KEY(username, login_time)
)
```

```
CREATE TABLE NOTIFICHE
(
    documento REFERENCES documenti(id_documento)
        ON DELETE CASCADE ON UPDATE CASCADE,
    dest_notifica REFERENCES impiegati(id_impiegato)
        ON DELETE CASCADE ON UPDATE CASCADE ,
    mitt_notifica REFERENCES impiegati(id_impiegato)
        ON DELETE CASCADE ON UPDATE CASCADE ,
    notifica VARCHAR2(25) NOT NULL ,
    CHECK(notifica IN ('in notifica', 'notificato', 'risposta alla notifica')),
    PRIMARY KEY(documento, dest_notifica, mitt_notifica)
)
```

INDICI

```
CREATE INDEX documenti$data  
ON documenti(data_ricezione)
```

```
CREATE INDEX documenti$mittente  
ON documenti(mittente)
```

```
CREATE INDEX documenti$ambito  
ON documenti(ambito)
```

```
CREATE INDEX documenti$responsabile  
ON documenti(responsabile)
```

```
CREATE INDEX destinatari$destinatario  
ON destinatari(destinatario)
```

```
CREATE INDEX cronologia$data_registrazione  
ON cronologia(data_registrazione)
```


OPERAZIONI

```
/* Trova il numero di documenti inviati da una persona */  
SELECT inviati  
FROM persone  
WHERE id_persona = ?
```

```
/* Trova il numero di documenti ricevuti da una persona */  
SELECT ricevuti  
FROM persone  
WHERE id_persona = ?
```

```
/* Ricerca i protocolli dei documenti per data / mittente / ambito */  
SELECT *  
FROM documenti  
WHERE data = ?  
OR mittente = ?  
OR ambito = ?
```

```
/* Ricerca i protocolli dei documenti per destinatario */  
SELECT *  
FROM documenti, destinatari  
WHERE id_documento = ?  
AND id_destinatario = ?
```

```
/* Ricerca i protocolli dei documenti per responsabile */  
SELECT *  
FROM documenti  
WHERE responsabile = ?
```

```
/* Ricerca dei documenti protocollati durante il giorno */  
SELECT *  
FROM documenti  
WHERE id_documento IN (  
    SELECT documento  
    FROM cronologia  
    WHERE data_registrazione = sysdate  
)
```

```
/* Ricerca della cronologia di un documento */  
SELECT *  
FROM cronologia  
WHERE documento = ?
```

PROCEDURE

```
CREATE PROCEDURE registra_documento( doc_registrazione CHAR(12),
                                     id_registratore CHAR(10) ,
                                     count_protocollo NUMBER(8)
                                   ) IS

BEGIN
  UPDATE documenti
  SET protocollo          = count_protocollo ,
      anno                = TO_CHAR(sysdate, 'YYYY'),
      stato               = 'registrato'
  WHERE documenti.id_documento = doc_registrazione;
  INSERT INTO cronologia (documento, data_registrazione, registratore )
  VALUES ( doc_registrazione ,
            sysdate           ,
            id_registratore    );
  COMMIT;
END registra_documento;
```

```
CREATE PROCEDURE smista_documento ( doc_notifica CHAR(12),
                                    mitt_notifica CHAR(10)
                                  ) IS

DECLARE
  CURSOR dest_cursor IS
    SELECT  destinatario
    FROM    destinatari
    WHERE   documento = doc_notifica;
  dest_variable dest_cursor %rowtype;
BEGIN
  OPEN dest_cursor;
  LOOP
    FETCH dest_cursor INTO dest_variable;
    EXIT WHEN dest_cursor %NOTFOUND;
    INSERT INTO notifiche VALUES ( documento ,
                                   mitt_notifica ,
                                   dest_variable.destinatario,
                                   'in notifica' );

  END LOOP;
  CLOSE dest_cursor;
  UPDATE cronologia
  SET smistatore          = mitt_notifica,
      data_smistamento    = sysdate
  WHERE cronologia.documento = doc_notifica;
  UPDATE documenti
  SET stato              = 'smistato'
  WHERE documenti.id_documento = doc_notifica;
  COMMIT;
END smista_documento;
```

```
CREATE PROCEDURE fascicola_documento ( doc_fascicolazione CHAR(12),
                                         id_fascicolatore CHAR(10),
                                         id_fascicolo CHAR(8)
                                         ) IS
BEGIN
    INSERT INTO fascicolati VALUES ( id_fascicolo,
                                      doc_fascicolazione
                                      ) ;

    UPDATE cronologia
    SET data_fascicolazione      = sysdate,
        fascicolatore           = id_fascicolatore
    WHERE cronologia.documento = doc_fascicolazione;
    UPDATE documenti
    SET stato                    = 'fascicolato'
    WHERE documenti.id_documento = doc_fascicolazione;
    COMMIT;
END fascicola_documento;
```

```
CREATE PROCEDURE login_event ( id_username VARCHAR2(20)
                               ) IS
BEGIN
    INSERT INTO logs(username, login_time) VALUES ( id_username ,
                                                    sysdate      ) ;

    COMMIT;
END login_event;
```

```
CREATE PROCEDURE logout_event ( id_username VARCHAR2(20)
                                ) IS
BEGIN
    UPDATE logs
    SET logout_time = sysdate
    WHERE logs.username = id_username
    AND logs.logout_time IS NULL;
    COMMIT;
END logout_event;
```

TRIGGER

```
/* Controllo sull'unicità di pt registratura per area*/
CREATE TRIGGER check_pt_registratura
BEFORE INSERT OR UPDATE OF pt_registratura ON uffici
FOR EACH ROW
DECLARE X NUMBER;
BEGIN
    SELECT SUM(pt_registratura) INTO X
    FROM uffici
    WHERE (area = :new.area) ;
    IF (X + :new.pt_registratura > 1)
    THEN raise_application_error( - 20001, 'Punto di registratura già esistente') ;
    END IF;
END;
```

```
/* Controllo sull'unicità del responsabile per ufficio*/
CREATE TRIGGER check_responsabile
BEFORE INSERT OR UPDATE OF responsabile ON impiegati
FOR EACH ROW
DECLARE X NUMBER;
BEGIN
    SELECT SUM(responsabile) INTO X
    FROM impiegati
    WHERE (ufficio = :new.ufficio) ;
    IF (X + :new.responsabile > 1)
    THEN raise_application_error( - 20002, 'Responsabile ufficio già esistente') ;
    END IF;
END;
```

```
/* Gestione del campo ricevuti di Persone */
CREATE TRIGGER count_ricevuti
AFTER INSERT OR DELETE OR UPDATE OF destinatario ON destinatari
FOR EACH ROW
BEGIN
    IF INSERTING THEN
        UPDATE persone SET ricevuti = ricevuti + 1
        WHERE ( id_persona = :new.destinatario ) ;
    END IF;
    IF DELETING THEN
        UPDATE persone SET ricevuti = ricevuti - 1
        WHERE ( id_persona = :old.destinatario ) ;
    END IF;
    IF UPDATING THEN
        UPDATE persone SET ricevuti = ricevuti - 1
        WHERE ( id_persona = :old.destinatario ) ;
        UPDATE persone SET ricevuti = ricevuti + 1
        WHERE ( id_persona = :new.destinatario ) ;
    END IF;
END;
```

```

/* Gestione del campo inviati di Persone */
CREATE TRIGGER count_inviati
AFTER INSERT OR DELETE OR UPDATE OF mittente ON documenti
FOR EACH ROW
BEGIN
    IF INSERTING THEN
        UPDATE persone SET inviati = inviati + 1
        WHERE ( id_persona = :new.mittente );
    END IF;
    IF DELETING THEN
        UPDATE persone SET inviati = inviati - 1
        WHERE ( id_persona = :old.mittente );
    END IF;
    IF UPDATING THEN
        UPDATE persone SET inviati = inviati - 1
        WHERE ( id_persona = :old.mittente );
        UPDATE persone SET inviati = inviati + 1
        WHERE ( id_persona = :new.mittente );
    END IF;
END;

```

```

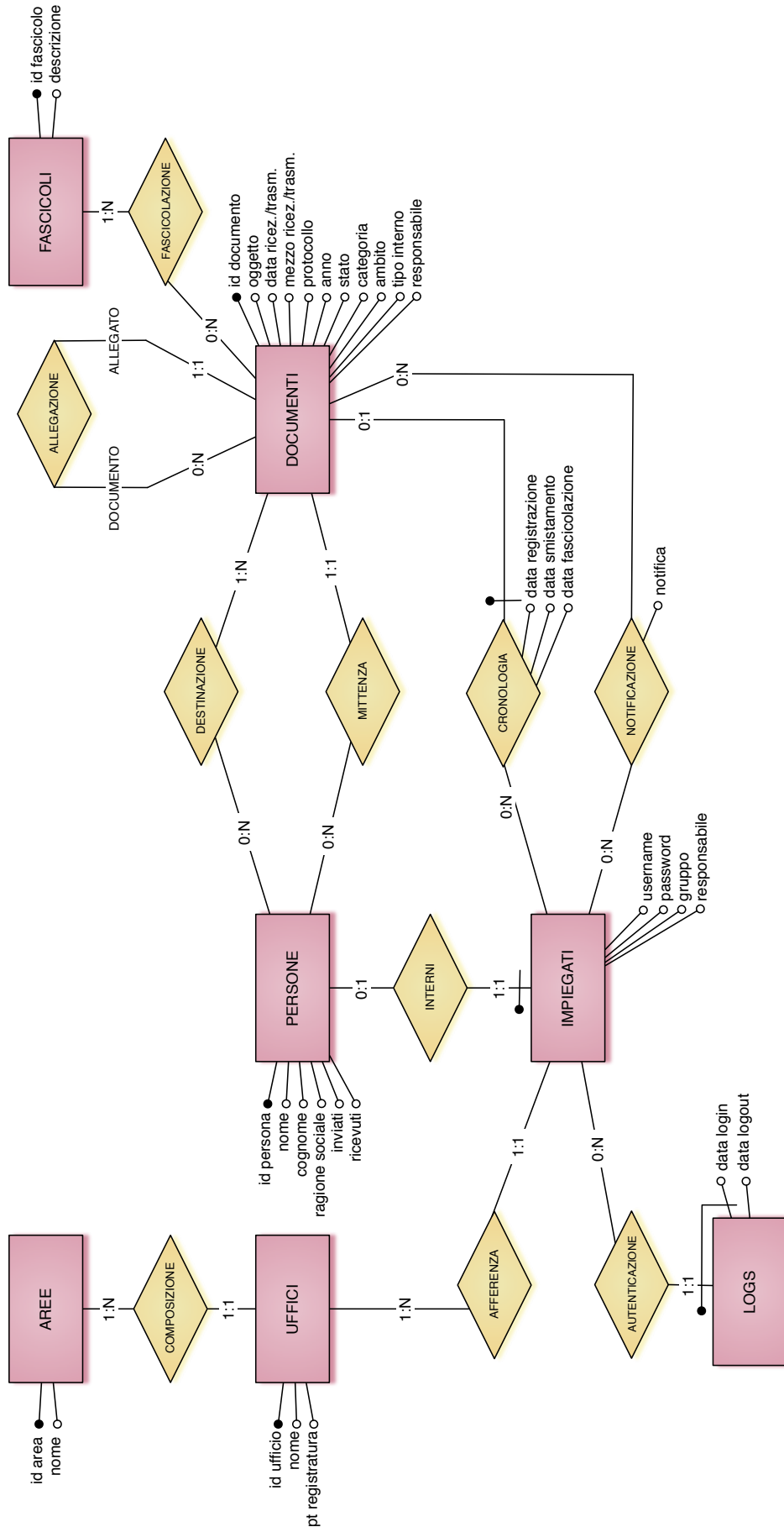
/* Controllo privilegi responsabile in Documenti */
CREATE TRIGGER check_responsabile_doc
BEFORE INSERT OR UPDATE OF responsabile ON documenti
FOR EACH ROW
DECLARE X char(1);
        irresponsabile EXCEPTION;
BEGIN
    SELECT responsabile INTO X
    FROM impiegati
    WHERE (id_impiegato = :new.responsabile);
    IF X = '0'
    THEN RAISE irresponsabile;
    END IF;
    EXCEPTION
    WHEN irresponsabile THEN
        RAISE_APPLICATION_ERROR
        (-20003,'Responsabile non è un id_impiegato con privilegi necessari');
END;

```

```

/* Gestione automatica della “Risposta alla notifica” e cancellazione. Abbiamo */
/* previsto che quando il documento passa allo stato notificato e il vecchio sta- */
/* to era ‘in notifica’ viene generata la ‘risposta alla notifica’ e la vecchia */
/* notifica viene cancellata. Se il nuovo stato è ‘notificato’ e quello vecchio */
/* era ‘risposta alla notifica’ allora si cancella la tupla e basta */
CREATE TRIGGER check_stato_notifica
AFTER UPDATE OF notifica ON notifiche
FOR EACH ROW
BEGIN
    IF (:new.notifica = 'notificato') AND (:old.notifica = 'in notifica') THEN
        INSERT INTO notifiche(documento, dest_notifica, mitt_notifica, notifica)
        VALUES(
            :old.documento,
            :old.mitt_notifica,
            :old.dest_notifica,
            'risposta alla notifica');
    END IF;
    DELETE FROM notifiche
    WHERE (:new.documento = :old.documento )
        AND (:new.mitt_notifica = :old.mitt_notifica)
        AND (:new.dest_notifica = :old.dest_notifica);
END;

```



POSSIBILI IMPLEMENTAZIONI

Si potrebbero applicare ulteriori controlli per la protezione e l'accesso ai dati, ad esempio:

- un trigger che blocchi *protocollo* e *anno*, una volta impostati, da un possibile aggiornamento;
- utilizzare una *sequence* per il numero di protocollo;
 - relativo evento che resettì la *sequence* il primo dell'anno;