

Descrizione della realtà in questione	2
Glossario	4
Schema scheletrico	6
Progettazione concettuale	7
Personale azienda	7
Mansioni, interventi, dispositivi di protezione e pericoli	11
Assistenze	17
Contenuti di documentazione	17
Schema ER	21
Progettazione Logica	22
Eliminazione gerarchie ISA	22
Selezione delle chiavi primarie ed eliminazione delle chiavi esterne	23
Trasformazione attributi composti o multipli	31
Traduzione di entità e associazioni in schemi e relazioni	32
Schema logico completo	40
Verifica della normalizzazione	42
Vincoli aggiuntivi	42
Studio dati derivati	43
Base di dati	46
Query di creazione	46
Triggers e Stored Procedure	53
Query di inserimento	59
Query di interrogazione	69
Query di modifica	72
Query di eliminazione	72
Bibliografia	74

Auxilium

Sistema di supporto per la task force di interventi Autostrade SPA

Descrizione della realtà in questione

Questo progetto nasce a seguito delle richieste del cliente Autostrade per l'Italia fornite durante l'evento Hackathon VarGroup 2018.

Si suppone di dover realizzare una base di dati per la memorizzazione delle informazioni utili relative all'organizzazione della task force di interventi sulle tratte operative gestite dal cliente. Con questo si intende di dover gestire in modo efficace:

- il personale aziendale con registro dei turni
- le mansioni, gli interventi, i dispositivi di protezione e i pericoli
- le assistenze effettuate dalla sede centrale agli operatori sul campo
- i contenuti di documentazione

La possibilità di reperire materiale documentativo prevede che un operatore dal proprio dispositivo sia capace di trovare informazioni utili alla mansione a lui assegnata e che sta svolgendo attraverso apposite domande o frasi. A questo proposito la base di dati memorizza i contenuti in una struttura ad albero ed inoltre deve essere tenuta traccia dei messaggi inviati all'operatore. L'individuazione della particolare risorsa è lasciata ad un'intelligenza artificiale che sfrutta un insieme di parametri per ricondursi alle frasi memorizzate e fornire una risposta adeguata all'utente finale. La struttura ad albero è strettamente necessaria perchè partendo da un concetto astratto, generico, che può essere la categoria di appartenenza di un gruppo di mansioni, conduce l'utente attraverso raffinamenti progressivi a concetti più specifici fino a raggiungere le "foglie" in cui sono descritte le mansioni. Per una determinata foglia possono anche essere eventualmente memorizzati documenti multimediali di supporto come video, immagini o pdf.

Con mansione si intendono le attività che possono essere svolte da un operatore di cui si memorizzano tutte le informazioni necessarie richieste, come descritto di seguito, al fine di garantire l'incolumità del lavoratore. È necessario tenere traccia delle condizioni di pericolo e dei dispositivi di sicurezza necessari e obbligatori che dovranno essere utilizzati nei determinati casi. I dispositivi di sicurezza a loro volta sono presi in prestito da un magazzino dove viene tenuta traccia della data di ritiro e reso.

Ciascun dipendente aziendale ha un registro dei turni di lavoro in cui, in base ad una specifica giornata lavorativa, viene specificata la tipologia di turno che dovrà essere da lui svolta. Le possibili tipologie di turno sono solamente due e sono le seguenti: *"Manutentore"* e *"Centralinista"*. Durante ogni turno è previsto che a ciascun dipendente sia assegnato univocamente un dispositivo digitale (DigitalCard) che consente di identificarlo nel sistema per tale giornata.

Un manutentore durante le sue giornate lavorative svolge gli interventi a lui assegnati ed appartenenti ad una mansione, per lo svolgimento di una mansione può anche essere richiesta la presenza di più operai che lavorano assieme.

Infine è necessario tenere traccia delle assistenze effettuate tra i manutentori e i centralinisti. Il centralinista è reperibile dagli operatori sul campo in qualunque momento della - *sua* - giornata lavorativa. Infatti, nel caso in cui questo primo livello di supporto basato su contenuti multimediali non bastasse, un manutentore può richiedere assistenza direttamente a un centralinista dislocato in sede.

Ipotesi aggiuntive

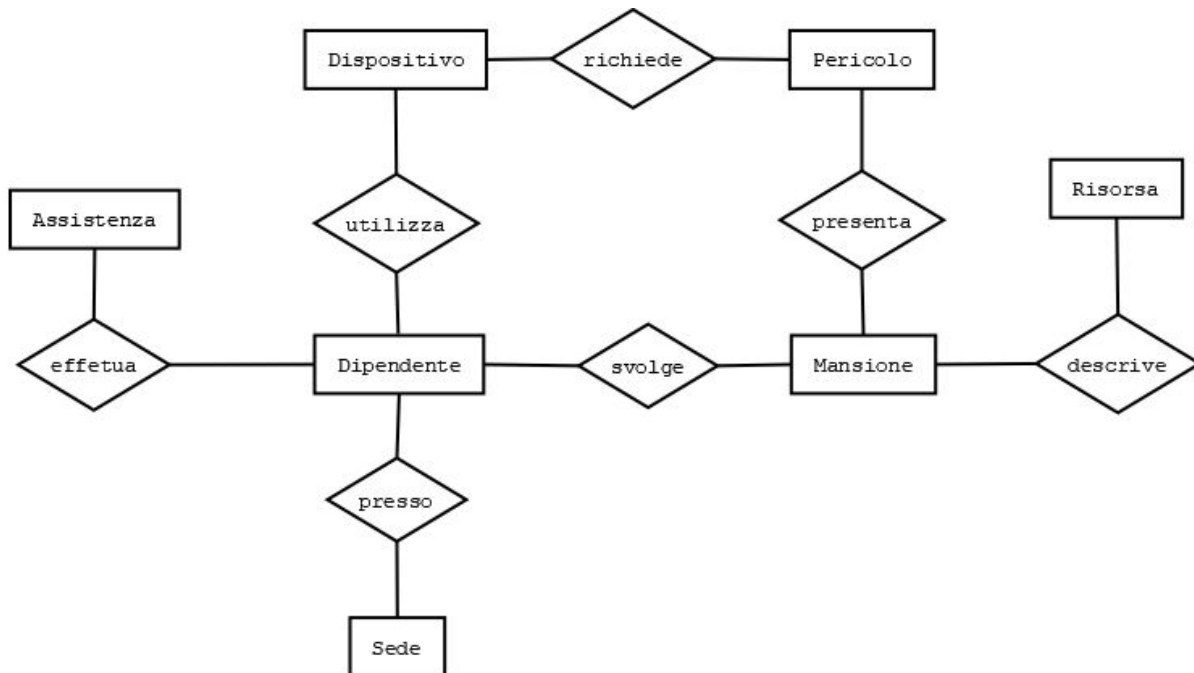
- Si trascura la necessità di dover gestire magazzini separati per reperire i dispositivi di protezione nelle varie sedi e che siano messi a disposizione sufficienti dispositivi per soddisfare le richieste di tutti i dipendenti.
- Per identificare una determinata zona geografica si farà uso del codice di avviamento postale (CAP) del comune della zona di riferimento e il campo indirizzo
- Gli indirizzi sono espresso nel formato: <Via, Civico>
- Per la gestione delle date e degli orari si farà uso dello standard ISO 8601 . Le date sono quindi espresse nel formato <YYYY-MM-DD> e gli orari in <hh:mm:ss>. Per quanto riguarda i timestamp la notazione utilizzata è la seguente <YYYY-MM-DD hh:mm:ss>

Glossario

Termine	Descrizione	Sinonimi	Legame
Dipendente	<ul style="list-style-type: none"> • CF • Nome • Cognome • Telefono • Mail • Indirizzo 	<ul style="list-style-type: none"> • Lavoratore • Utente 	<ul style="list-style-type: none"> • Sede • Turno
Turno	<ul style="list-style-type: none"> • Data • Inizio • Fine • DigitalCard 	<ul style="list-style-type: none"> • Giornata lavorativa 	<ul style="list-style-type: none"> • Dipendente • Centralinista • Manutentore
Centralinista		<ul style="list-style-type: none"> • Operatore in sede • Operatore di supporto • Operatore specializzato • Assistente remoto 	<ul style="list-style-type: none"> • Assistenza • Turno
Manutentore		<ul style="list-style-type: none"> • Operaio • Operatore sul campo • Tecnico 	<ul style="list-style-type: none"> • Mansione • Dispositivo • Assistenza • Turno
Assistenza	<ul style="list-style-type: none"> • Inizio • Fine • Ticket 	<ul style="list-style-type: none"> • Videochiamata remota • Assistenza virtuale 	<ul style="list-style-type: none"> • Manutentore • Centralinista
Risorsa	<ul style="list-style-type: none"> • Titolo • Codice 	<ul style="list-style-type: none"> • Documento 	<ul style="list-style-type: none"> • Mansione • Media
Media	<ul style="list-style-type: none"> • Identificativo • Posizione • Tipologia 	<ul style="list-style-type: none"> • Contenuto multimediale • Allegato 	<ul style="list-style-type: none"> • Risorsa
Condizione	<ul style="list-style-type: none"> • Nome 	<ul style="list-style-type: none"> • Pericolo 	<ul style="list-style-type: none"> • Gruppo
Dispositivo	<ul style="list-style-type: none"> • Nome • Caratteristiche 	<ul style="list-style-type: none"> • Dispositivo di sicurezza • Dispositivo di protezione individuale (DPI) 	<ul style="list-style-type: none"> • Gruppo • Manutentore

Gruppo	<ul style="list-style-type: none">• Titolo• Pericolo	<ul style="list-style-type: none">• Insieme di pericoli e dispositivi di protezione	<ul style="list-style-type: none">• Condizione• Dispositivo• Risorsa
Mansione	<ul style="list-style-type: none">• Nome• Descrizione• Inizio• Fine• Indirizzo	<ul style="list-style-type: none">• Task• Intervento	<ul style="list-style-type: none">• Manutentore
Sede	<ul style="list-style-type: none">• Codice• Telefono• Mail• Indirizzo	<ul style="list-style-type: none">• Centrale	<ul style="list-style-type: none">• Dipendente

Schema scheletrico



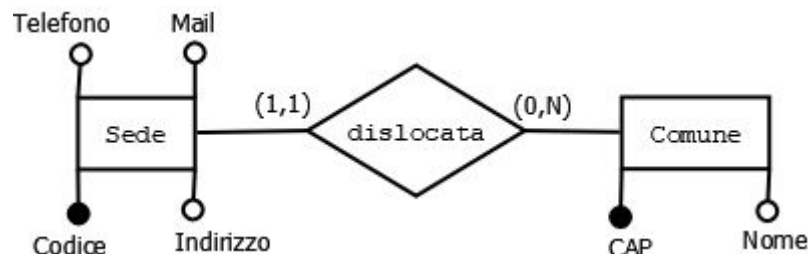
Progettazione concettuale

Le specifiche sugli attributi caratterizzanti per ogni entità e le informazioni progettuali che hanno portato alla costruzione passo-passo dello schema ER sono trattate in questa sezione. A livello di metodologia di progettazione si è fatto uso di un approccio mixed che ha permesso di suddividere il progetto in tanti concetti base (bottom-up) successivamente analizzati nel dettaglio tramite una metodologia (top-down) e di seguito riuniti tra loro nello schema finale.

Personale azienda

Sede, Comune

Per far fronte alla necessità di gestire i luoghi si è deciso di introdurre l'entità *Comune* rappresentativa di un territorio sullo stato italiano.



Proprietà e Relazione

Comune

L'entità *Comune* è stata introdotta per identificare a livello geografico i territori presso i quali, come si vedrà di seguito, sono dislocate le sedi, gli interventi e i dipendenti. *Comune* è descritto dal valore Nome e dal CAP - *codice di avviamento postale* - utilizzato anche come chiave per la sua proprietà di unicità.

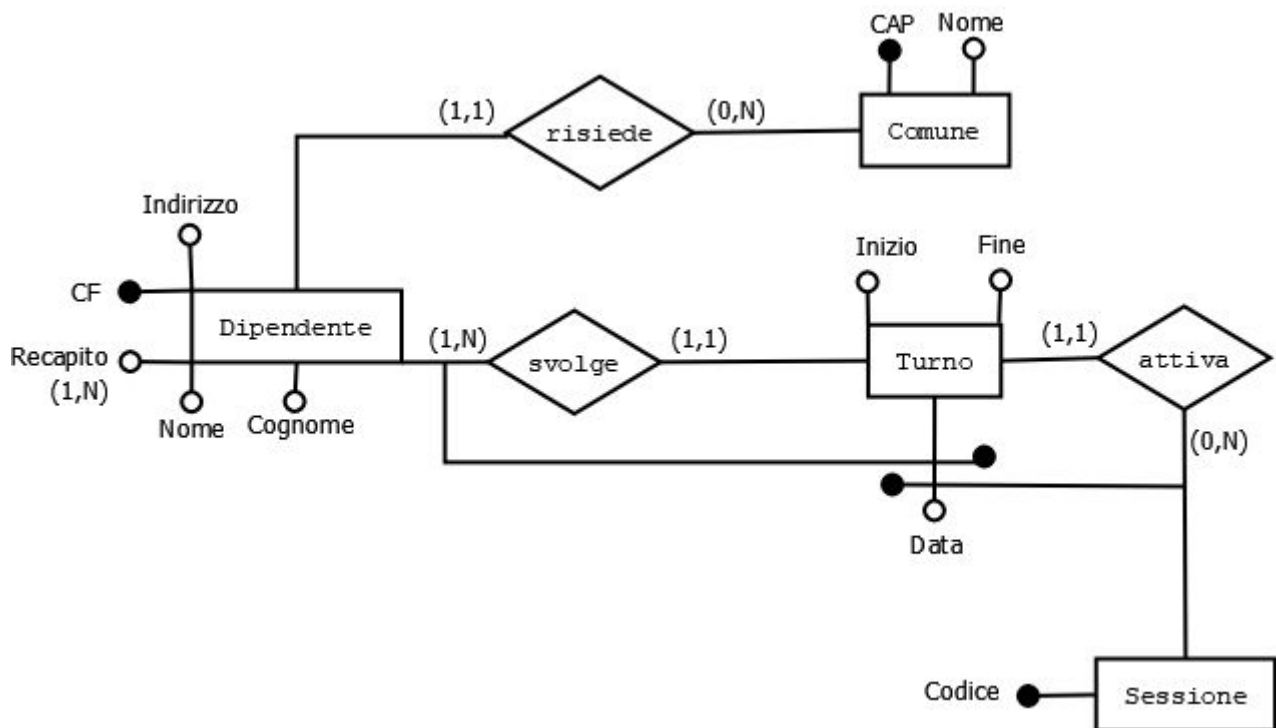
Sede

L'entità *Sede* rappresenta il centro operativo di una determinata zona geografica presso la quale i dipendenti in prossimità della stessa fanno riferimento. *Sede* è descritta dai seguenti attributi: Telefono, Mail, Indirizzo e un Codice utilizzato anche come valore identificativo.

Le entità *Sede* e *Comune* sono in relazione di tipo uno a molti tra loro mediante "*dislocata*" per completare il valore espresso da Indirizzo al fine di identificare geograficamente la posizione di una *Sede* senza possibilità di equivoco.

Dipendente, turno e sessione

"[...] Ciascun dipendente aziendale ha un registro dei turni di lavoro in cui, in base ad una specifica giornata lavorativa, viene specificata la tipologia di turno che dovrà essere da lui svolta [...]"



Proprietà e Relazione

Dipendente

Con *Dipendente* si intende esprimere il concetto di un membro aziendale che fa riferimento ad una determinata *Sede*. *Dipendente* è descritto da alcuni dati anagrafici (CF, Nome, Cognome), un indirizzo - di residenza - e da recapiti telefonici che possono essere più di uno. Data la sua caratteristica di univocità si è deciso di utilizzare il codice fiscale anche come valore di chiave. Il luogo di residenza di un dipendente è dato dal valore dell'indirizzo e dal comune di residenza indicato dalla relazione "*risiede*".

Un *Dipendente* fa riferimento ad una sola determinata *Sede* mentre ad una *Sede* fanno riferimento da uno a molti *Dipendenti*. Come specificato dalle richieste del cliente, con questo schema non si intende fornire una descrizione accurata della gestione aziendale ma bensì rappresentare le informazioni indispensabili per tracciare dipendenti nelle rispettive sedi per la gestione della task force di interventi.

Sessione

Con *Sessione* si intende una sorta di dispositivo digitale - *DigitalCard* - contraddistinto da un valore univoco Codice utilizzato anche come chiave.

Turno

Con *Turno* si intende invece gestire e tenere traccia dell'organizzazione a livello lavorativo dei vari *Dipendenti* nel corso del tempo. Il concetto di turno deve essere inteso come un lasso di tempo di

operatività di un *Dipendente* durante il quale quest'ultimo può essere chiamato a svolgere determinati "compiti".

Essendo i dipendenti membri di una task force di interventi è chiaro che non si possono programmare le attività in anticipo dato che non è facile prevedere la maggior parte degli interventi che saranno emergenze - *escluse le manutenzioni programmate* - e che nella maggior parte dei casi i turni sono flessibili e non seguono uno schema ben preciso che può quindi variare da giornata in giornata.

Un turno è contraddistinto dalle seguenti proprietà: la data in cui quest'ultimo verrà svolto e da due attributi Inizio e Fine per indicare rispettivamente l'orario di inizio e di fine del turno in questione.

Turno è in relazione con *Dipendente* tramite "svolge" per esprimere i turni svolti da un determinato dipendente, questa relazione è di tipo uno a molti dato che un turno lavorativo è svolto da un solo dipendente e allo stesso tempo un dipendente svolge da uno a più turni di lavoro. La chiave di turno è quella composta tra il codice fiscale del dipendente (CF) e la data di svolgimento del turno per esprimere il vincolo di legame tra le due entità.

"[...] Durante ogni turno è previsto che a ciascun dipendente sia assegnato univocamente un dispositivo digitale che consente di identificarlo nel sistema per tale giornata [...]".

Per esprimere questo concetto, *Turno* è in relazione con *Sessione* tramite "attiva". Il vincolo è reso esplicito dalla seconda chiave di Turno composta dal valore *Data* dello stesso e il codice della sessione; in questo modo viene espresso che durante quel turno verrà utilizzato uno ed un solo dispositivo identificativo digitale e anche che quel dispositivo, durante quel turno, non potrà essere riassegnato.

Ulteriori considerazioni

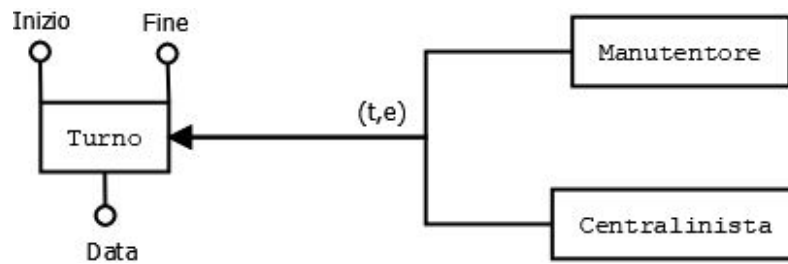
Assegnando il codice di sessione al dipendente in base al turno piuttosto che alla persona stessa si dà la possibilità al sistema di poter riutilizzare tale valore in giornate differenti a seconda dei turni. In questo modo non sarà necessario dover gestire un codice di sessione per il numero totale di dipendenti ma solo per quelli che effettivamente lavorano nella giornata, il sistema inoltre risulterà più efficiente e flessibile a gestire casi in cui tale dipendente non contribuirà più alla task force di interventi.

Centralinista e Manutentore

È necessario gestire 2 figure distinte:

- *Centralinista* che rappresenta un operaio specializzato dislocato nella propria sede di riferimento e che si mette a disposizione per assistenze da remoto.
- *Manutentore* che effettua interventi sul campo nell'arco della giornata lavorativa e che può effettuare un'assistenza con i centralinisti.

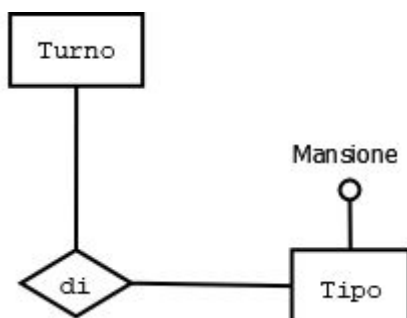
A tal proposito si è pensato di generalizzare il concetto di *Turno* e specializzare *Centralinista* e *Manutentore*. È stato scelto appositamente *Turno* - e non *Dipendente* - poiché si vuole garantire la possibilità di cambi di ruolo nel corso del tempo.



Proprietà e Relazione

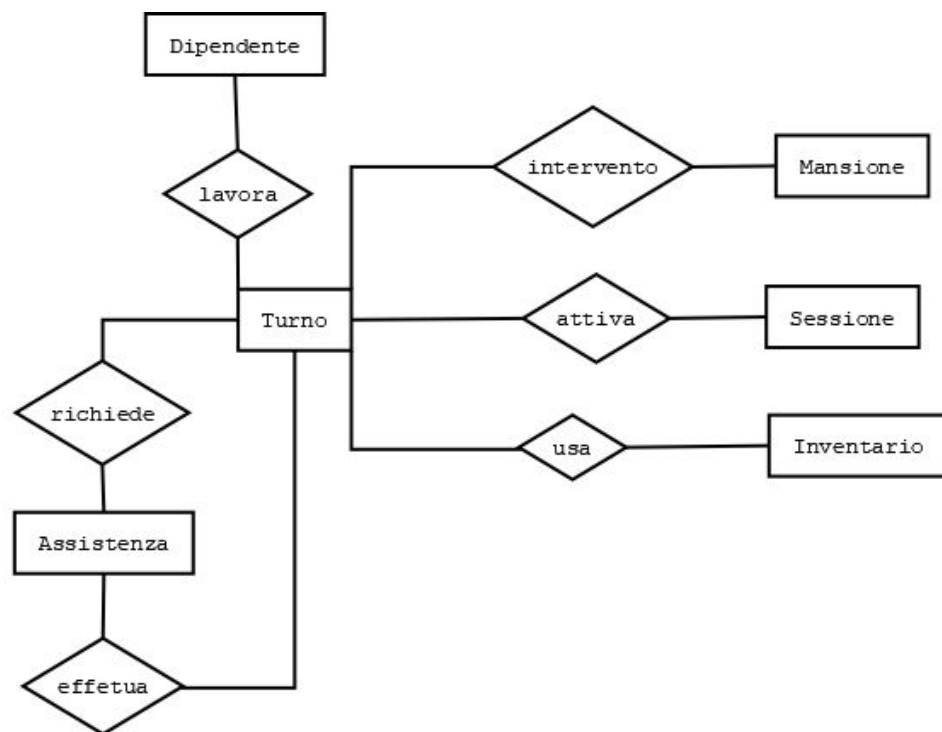
La generalizzazione prevede che vengano gestiti solo turni di tipo *manutentore* o *centralinista*. In base alle specifiche non sono contemplate ulteriori tipologie di turno che si intende rappresentare.

Ulteriori considerazioni



Considerando che nella soluzione precedentemente proposta le entità specializzate presentano le stesse proprietà si può pensare ad una schema più semplice come quello proposto a lato. In questo modo la base di dati risulterebbe più flessibile dando la possibilità di inserire più specializzazioni di tipo ruolo rispetto alle sole *centralinista* e *manutentore* della versione originale.

Sebbene la soluzione alternativa può sembra più funzionale, questa variante ha lo svantaggio di non fare chiarezza in termini di controllo sulla qualità dei dati: l'entità turno verrebbe sovraccaricata con le relazione di entrambe le entità specializzate causando, di conseguenza, la presenza di relazioni di *Manutentore* come "*usa*" o "*intervento*" anche per *Centralinista* e viceversa per quanto riguarda il discorso delle assistenze. Questo difetto andrebbe corretto successivamente.

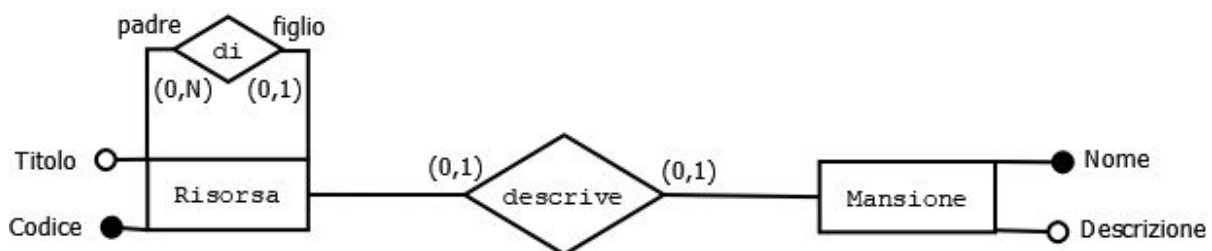


Sopra è riportato un ipotetico schema scheletrico semplificato realizzato adottando la variante “*Tipo di Turno*”. È molto facile notare la confusione che si verrebbe a formare, proprio per questo motivo delle due soluzioni si preferisce e si adotta la prima.

Mansioni, interventi, dispositivi di protezione e pericoli

Mansione

“[...] Con mansione si intendono le attività che possono essere svolte da un operatore di cui si memorizzano tutte le informazioni necessarie con tanto di pericoli e dispositivi di protezione al fine di garantire l’incolumità del lavoratore [...]”



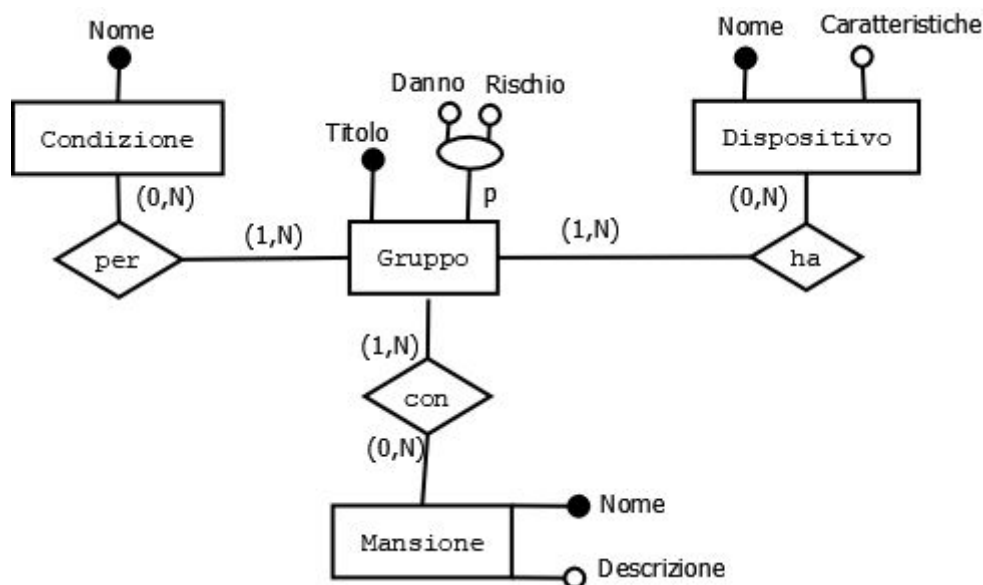
Proprietà e Relazione

Gli attributi di *Mansione* sono: Nome e Descrizione. Il primo è un valore non ambiguo e univoco che rappresenta la mansione e viene fornito dalla documentazione aziendale del cliente mentre il secondo, descrizione, è un attributo generico in cui vengono inserite delle informazioni aziendali relative alla stessa mansione. L’attributo Nome è anche utilizzato come chiave.

Una particolare mansione può essere documentata, ovvero, essere reperibile all'interno dell'albero delle risorse in cui viene descritta in un modo più approfondito e con aggiunta di eventuale materiale multimediale di supporto accessibile dall'operatore sul campo tramite il sistema di "assistenza automatizzata (bot)". Questo legame è espresso tramite la relazione "describe", una risorsa può eventualmente fare riferimento ad una mansione e viceversa una mansione può essere descritta da una risorsa.

Gruppo (dispositivi di protezione e pericoli)

Particolarità di questo sistema informativo è la capacità di memorizzare le informazioni riguardanti la pericolosità di una mansione che dovrà essere eseguita da un manutentore. Per realizzare tale ideologia si è fatto uso di un concetto astratto definito *gruppo* il quale serve a descrivere casi in cui pericoli distinti si concretizzano in una situazione di rischio precisa e che necessita di un insieme di dispositivi di protezione per essere affrontata in sicurezza da parte del manutentore.



Proprietà e Relazione

Condizione

Con *Condizione* si intende esprimere una condizione base che in concomitanza di altre condizioni costituisce un pericolo - *definito da Gruppo* -. *Condizione* è descritta dalla sola proprietà univoca **Nome** che viene utilizzata anche come chiave.

Dispositivo

Con *Dispositivo* si intende esprimere un dispositivo di protezione individuale (DPI) che dovrà essere utilizzato da parte di un manutentore incaricato di svolgere una mansione al fine di lavorare in modo sicuro. *Dispositivo* è descritta dalla proprietà univoca **Nome** che viene utilizzata anche come chiave e delle **Caratteristiche** da intendere come una descrizione delle specifiche tecniche.

Gruppo

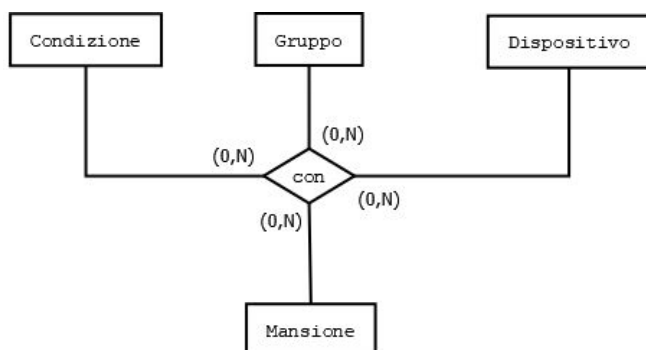
Con *Gruppo* si intende descrivere un insieme - *gruppo* - di condizioni che si possono verificare e che in concomitanza tra di loro costituiscono un pericolo per l'operatore finale incaricato di svolgere una mansione, per tale gruppo vengono inoltre riportati un insieme di dispositivi di protezione. Per insieme si intende che per esistere debba essere composto da almeno un elemento.

Gruppo è descritto da un Titolo univoco identificativo utilizzato anche come chiave e da un attributo composto *p* significativo di pericolo espresso dagli attributi Danno e Rischio su una scala di valori numerici con estremi rispettivamente 1 - *minimo* - e 3 - *massimo* -.

Attraverso la relazione N:M *per* tra le entità *Gruppo* e *Condizione* si riesce ad esprimere il grado di appartenenza di una condizione ad un gruppo, viceversa tale condizione può appartenere a gruppi differenti.

Infine la relazione, sempre di tipo N:M, *ha* tra le entità *Gruppo* e *Dispositivo* esprime il grado di appartenenza di un dispositivo di protezione ad un gruppo, viceversa tale dispositivo può appartenere a gruppi differenti.

Ulteriori considerazioni



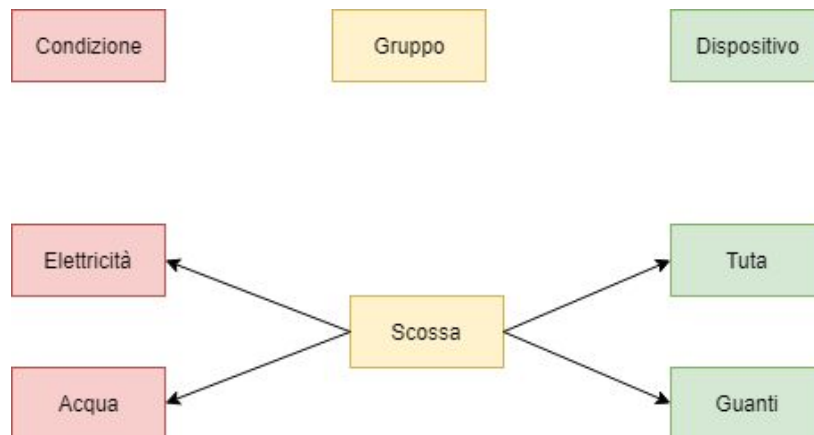
Una variante per rappresentare questo concetto è quella di utilizzare un associazione quaternaria tra le entità *Condizione*, *Gruppo*, *Dispositivo* e *Mansione*. Ad un primo impatto questa soluzione può sembrare più immediata ed intuitiva ma verrebbe sicuramente a mancare il concetto di *gruppo* inteso come insieme per condizioni e dispositivi.

Si adotta quindi la prima versione - iniziale - per il fatto che *Gruppo* deve essere trattato come entità separata al fine di ottenere i seguenti vantaggi:

- nel caso due *Mansioni* dipendono dagli stessi gruppi - o da una parte di essi - riduce notevolmente la ridondanza con la possibilità di riutilizzare quelli già esistenti
- impedisce che venga perso il significato di "*gruppo*" che, nella variante proposta, potrebbe essere mal interpretato e complicherebbe le fasi di progetto successive.

Esempio

Sono riportati alcuni casi di esempio al fine di comprendere meglio quanto descritto dalla relazione mentre successivamente verranno proposti casi reali estrapolati dalla documentazione fornita dal cliente per mansioni riguardanti il componente gruppo elettrogeno.



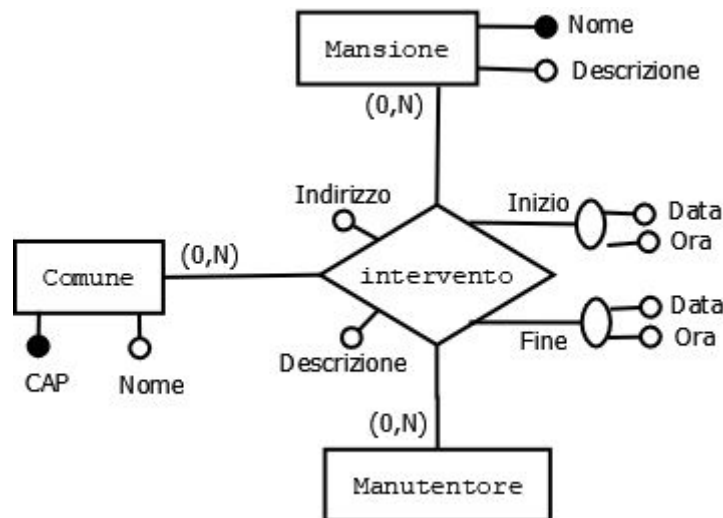
Condizioni		Dispositivi		M1	Mansioni
C1	Puntali protetti	D1	Puntali		prova sotto carico della apparecchiatura (con carico) per un tempo di 30 minuti e rilevazione grandezze elettriche
C2	Puntali scoperti	D2	Guanti		
C3	Parti scoperte in tensione entro 30 cm	D3	Maschera facciale		
C4	Parti scoperte in tensione a distanza	D4	Indumenti a norma		
C5	Parti scoperte in qualsiasi condizione				

Condizione	Gruppo	Dispositivo
C1	<u>G1</u>	D1
C4		
C1	<u>G2</u>	D1
C3		D2
C2	<u>G3</u>	D1
C5		D2
		D4

Gruppo	Mansione
<u>G1</u>	M1
<u>G2</u>	M1
<u>G3</u>	M1

Intervento

“[...] Un manutentore durante le sue giornate lavorative svolge gli interventi a lui assegnati ed appartenenti ad una mansione. [...]”. L'intervento descrive quindi l'effettivo svolgimento di una mansione da parte di uno o più operatori sul campo.



Proprietà e Relazione

Intervento viene rappresentato come una relazione ternaria tra *Comune*, *Manutentore* e *Mansione* ed è caratterizzato dalle seguenti proprietà:

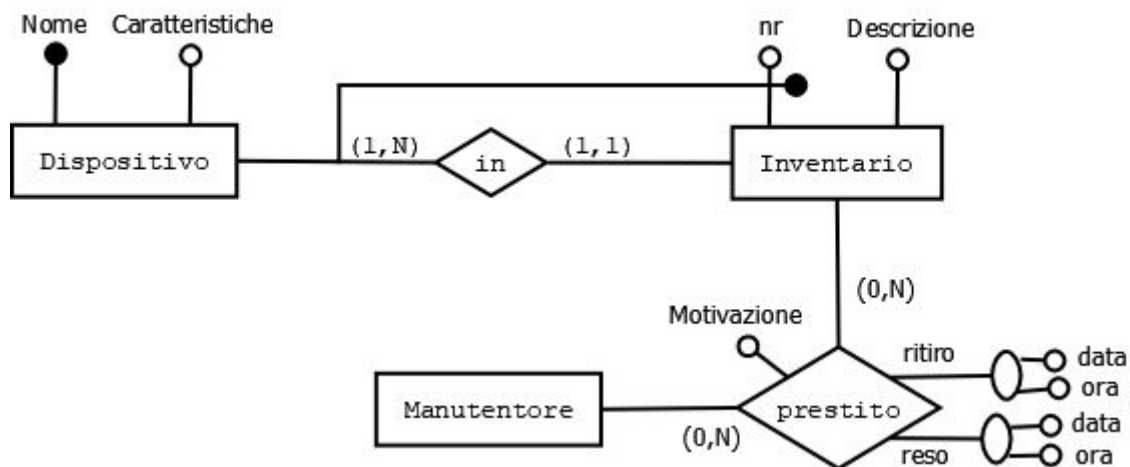
- Indirizzo per esprimere la posizione geografica complementariamente a Comune
- Descrizione dell'intervento svolto con la possibilità di aggiungere ulteriori informazioni
- Inizio come attributo composto per esprimere la Data e Ora di inizio Intervento
- Fine come attributo composto per esprimere la Data e Ora di fine Intervento

Di seguito sono riportati i vincoli che verranno trattati in modo approfondito nella progettazione logica assieme ai vincoli di chiave:

1. un manutentore non può fare altri interventi se ne ha già altri in corso di svolgimento
2. non ci possono essere più interventi dello stesso manutentore sovrapposti a livello temporale.

Inventario dispositivi e prestito

Per la gestione dell'utilizzo dei dispositivi di protezione individuale durante le giornate lavorative dei vari dipendenti si farà uso del concetto di inventario dove gli stessi verranno conservati e identificati per poter poi essere assegnati ai dipendenti quando questi ultimi lo richiedono. Il sistema richiede esplicitamente che venga memorizzato quando viene ritirato e reso un dispositivo da parte di un dipendente tracciando uno storico di utilizzo e che uno stesso dispositivo non venga assegnato a più dipendenti nello stesso lasso di tempo



Proprietà e Relazione

Inventario

Con *Inventario* si intende esprimere il concetto di magazzino presso il quale i dispositivi vengono intesi come copie fisiche presenti e utilizzabili da parte degli operatori.

Un oggetto presente nell'inventario è caratterizzato dal numero progressivo di copia dello stesso (attributo nr) e da un campo Descrizione in cui vengono inserite ulteriori informazioni aggiuntive. La chiave è difatti quella composta tra la chiave esterna di Dispositivo (Nome) e il valore di copia (nr).

Il legame di appartenenza di un oggetto nell'inventario ad una classe di dispositivo è espresso mediante la relazione "in" di tipo (1,N) e con le seguenti caratteristiche, ad un dispositivo possono corrispondere da una a più copie dello stesso nell'inventario e viceversa un elemento nell'inventario fa riferimento ad uno ed un solo dispositivo.

Prestito

Il concetto di utilizzo (prestito) di un dispositivo messo a disposizione - *al fine di svolgere gli interventi assegnati in sicurezza* - e reperibile dal *Manutentore* presso la sede di riferimento è reso esplicito dall'associazione "prestito". La relazione è di tipo N:M dato che ad un manutentore può essere prestato più volte lo stesso oggetto e allo stesso tempo possono essere richiesti più di un dispositivo di protezione alla volta. Le proprietà che caratterizzano questa relazione sono le seguenti:

- Motivazione per inserire eventuali informazioni aggiuntive riguardanti il ritiro
- Ritiro come attributo composto per esprimere la Data e Ora di ritiro del dispositivo
- Ritiro come attributo composto per esprimere la Data e Ora di restituzione del dispositivo

Si considerino inoltre i seguenti vincoli che verranno descritti accuratamente nella fase di progettazione logica assieme alla gestione delle chiavi:

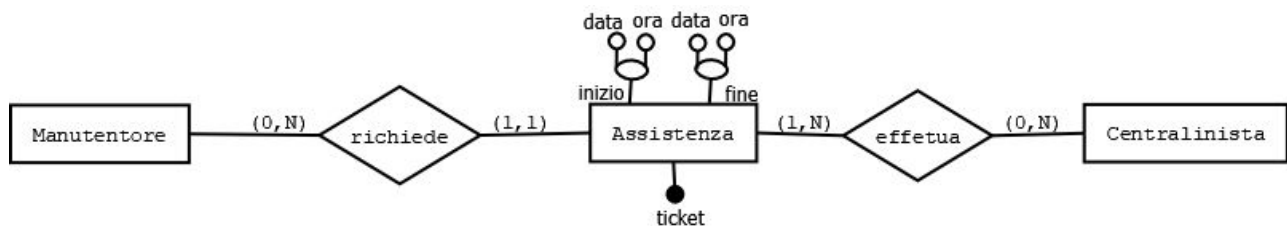
1. possono essere prelevati più oggetti - *fisicamente differenti* - dello stesso tipo in una stessa giornata lavorativa e non

2. non può essere assegnato uno stesso oggetto a più dipendenti in uno stesso lasso di tempo

Assistenze

Assistenza

“[...] Durante le giornate lavorative è necessario tenere traccia delle assistenze effettuate tra i manutentori e i centralinisti. Il centralinista è reperibile dagli operatori sul campo in qualunque momento della - sua - giornata lavorativa [...]”.



Proprietà e Relazione

Con assistenza si intende quindi rappresentare il concetto di *videochiamata remota* che un *Manutentore* può richiedere durante la sua giornata lavorativa e che verrà successivamente gestita da uno o più centralinisti *Centralinista* dislocati nelle relative sedi di riferimento.

Gli attributi descrittivi di tale entità sono dunque i seguenti:

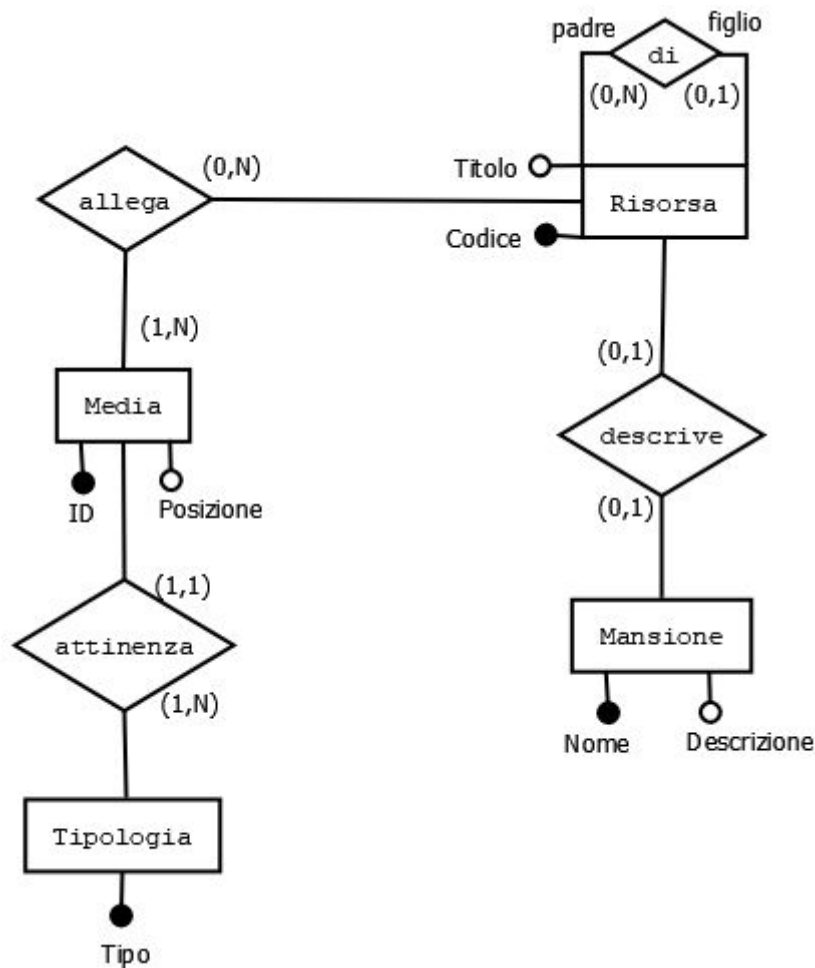
- ticket come valore numerico progressivo delle richieste di assistenza utilizzato anche come chiave
- Inizio come attributo composto per esprimere la Data e Ora di inizio dell'assistenza
- Fine come attributo composto per esprimere la Data e Ora di fine dell'assistenza

Assistenza è in relazione di tipo *1:N* con *Manutentore*, ovvero l'operatore che durante la sua giornata lavorativa può eventualmente richiedere assistenze remote, e con *Centralinista* tramite la relazione di tipo *N:M* "*effettua*" dato che un centralinista può contemporaneamente essere collegato su più assistenze e allo stesso tempo ad una assistenza possono partecipare più centralinisti. Un'assistenza è caratterizzata dalla presenza di un manutentore - *da cui la cardinalità minima 1* - e da uno a più centralinisti.

Contenuti di documentazione

Risorsa, Media, Tipologia

Con risorsa si intende un qualsiasi contenuto informativo o documentativo riguardante le mansioni di un operatore e non. Si è dovuto realizzare uno schema che funzionasse come dataset di supporto per tutto ciò che riguarda il sistema di intelligenza artificiale intermedio, il quale, permette di identificare una risorsa a partire da un insieme di caratteri testuali e che successivamente verrà inoltrata come risposta all'utente.



Proprietà e Relazione

La risorsa è identificata da un codice e un titolo rappresentativo. È proprio quest'ultimo valore testuale che verrà visualizzato in prima battuta come risposta e allo stesso tempo è la prima base di appoggio che permette all'intelligenza artificiali di effettuare il matching sulle frasi in input. Il sistema di text-matching verrà a sua volta integrato con ulteriori valori quali la struttura dell'albero, un report sui messaggi passati ed infine tutte le informazioni reperibili nel dettaglio mansione.

Tipologia

Con *Tipologia* si intende definire tutte le possibili varianti di contenuto multimediale che possono essere messe a disposizione. Un esempio comune di valori sono: “pdf”, “video”, “image”, definiti dall’attributo Tipo utilizzato anche come valore di chiave.

Media

Un *Media* rappresenta una risorsa caricata sul server e visualizzabile da parte degli utenti. Media è descritto dal valore testuale Posizione - *simile ad un URL* - che permette di reperire il *"file"* sul server centrale presso il quale è memorizzato e da un valore identificativo progressivo utilizzato

come chiave. La relazione di tipo uno a molti esprime il legame di appartenenza di una risorsa multimediale a una determinata tipologia.

Risorsa

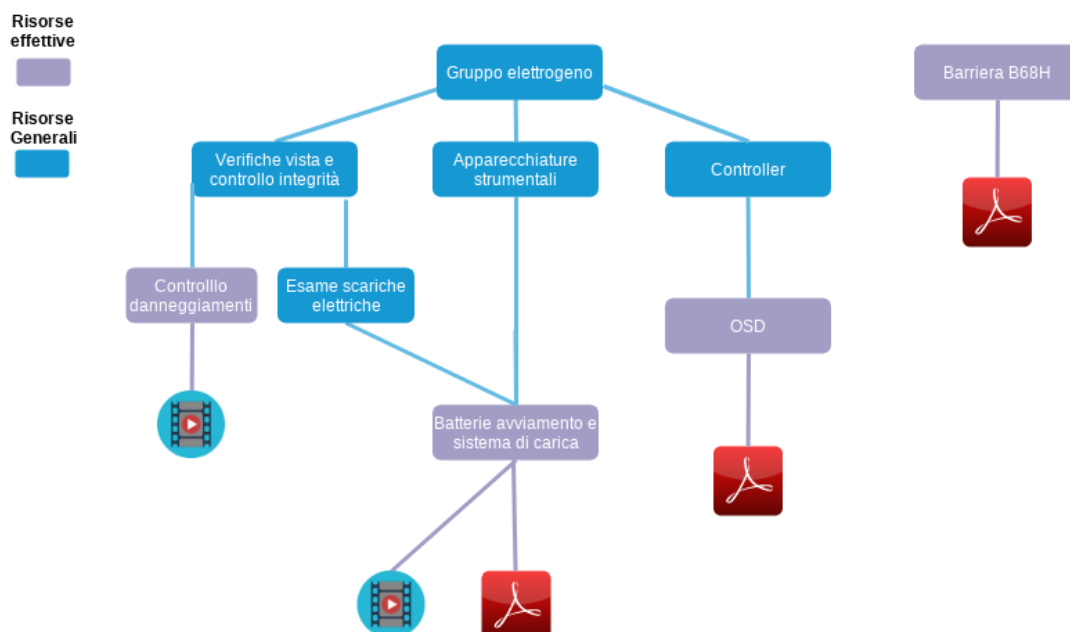
“[...] La struttura ad albero è strettamente necessaria perchè partendo da un concetto astratto, generico, che può essere la categoria di appartenenza di un gruppo di mansioni, conduce l'utente attraverso affinamenti progressivi a concetti più specifici fino a raggiungere le foglie in cui sono descritte le mansioni [...]”.

Ciò significa che al seguito di una richiesta si possono direttamente trovare i collegamenti ad altre risorse organizzate tra loro in modo gerarchico, grazie alla peculiarità della struttura ad albero. Quest'ultima è realizzata mediante l'auto-associazione “di” in cui viene esplicitato il grado di parentela di una risorsa figlia nei confronti di quella padre. Una risorsa ha al più un solo padre - *nessuno nel caso di radice* - quindi cardinalità massima 1 e allo stesso tempo una risorsa può avere da nessuno - *nel caso di foglia* - a più figli.

Media e Risorsa sono in relazione tra loro tramite la relazione di tipo molti a molti “allega” per indicare che a una data risorsa possono corrispondere più contenuti mediatici - *non necessariamente dello stesso tipo* - e che viceversa un media può essere allegato in risposta da più risorse. La cardinalità minima 0 serve per esprimere un grado di opzionalità in quanto non necessariamente ad una risorsa devono corrispondere media.

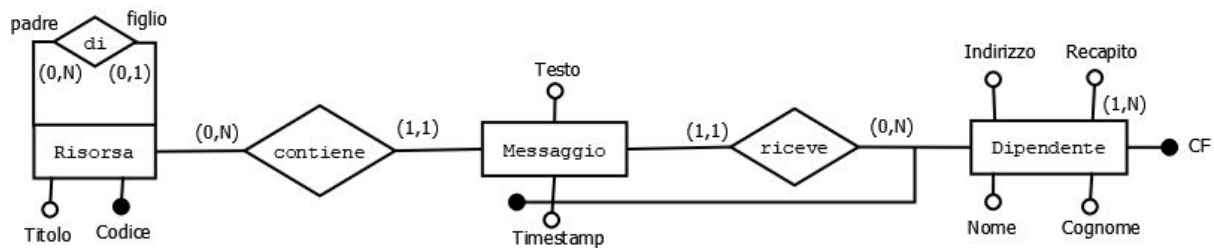
Esempio

In seguito è riportato un esempio che illustra graficamente una possibile struttura ad albero basata su un piccolo dataset di informazioni aziendali. Questo albero può essere tradotto seguendo il formato della relazione precedentemente descritta per rappresentare in modo equivalente queste informazioni all'interno della base di dati.



Messaggio

“[...] La possibilità di reperire materiale documentativo prevede che un operatore dal proprio dispositivo sia capace di trovare informazioni utili alla mansione a lui assegnata e che sta svolgendo attraverso apposite domande o frasi. A questo proposito la base di dati memorizza i contenuti in una struttura ad albero ed inoltre deve essere tenuta traccia dei messaggi inviati all'operatore [...]”.

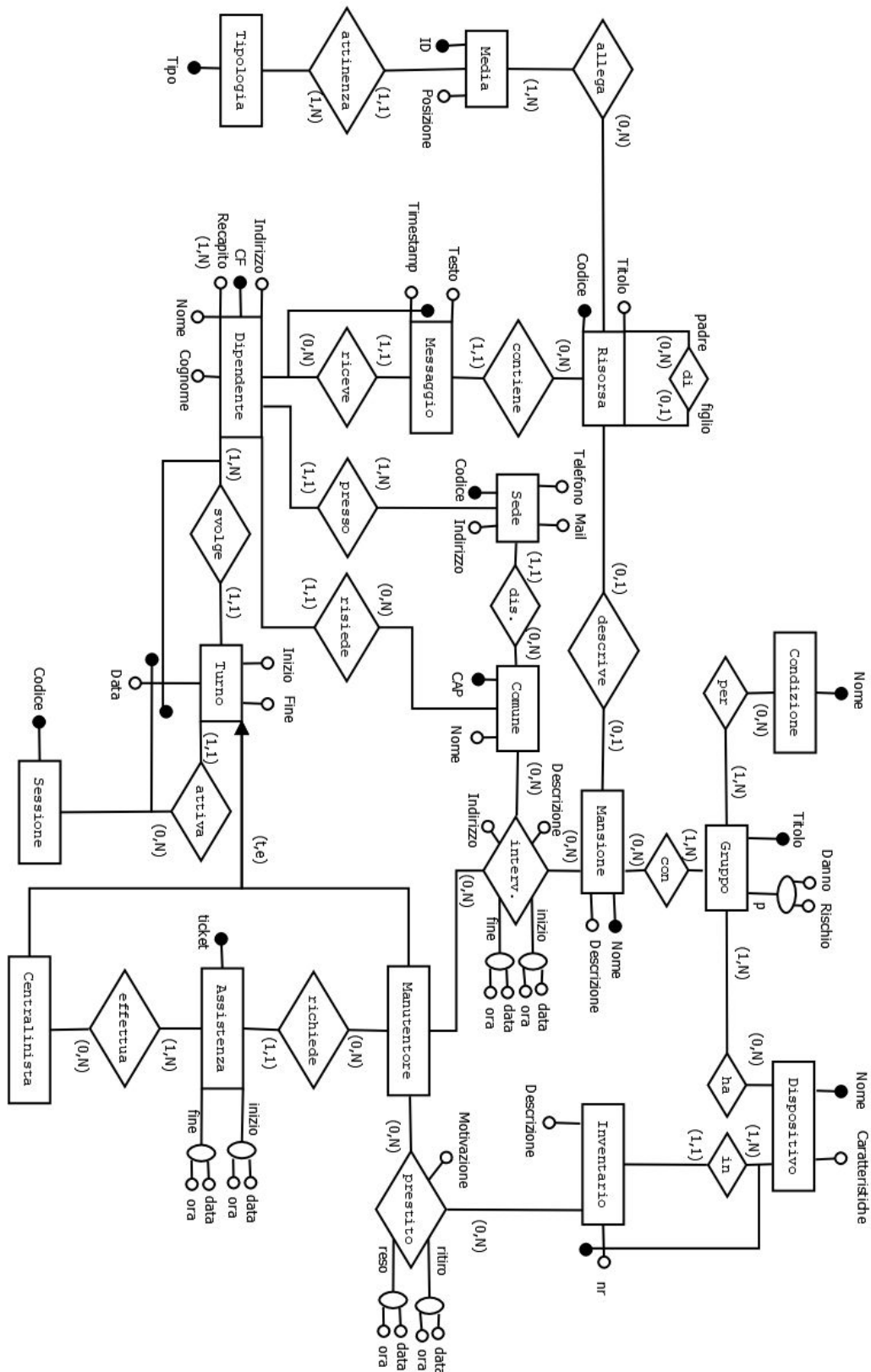


Proprietà e Relazione

Si intende gestire i messaggi ricevuti da un dipendente a seguito di una sua richiesta al sistema. Le proprietà caratterizzanti l'entità *Messaggio* sono: Testo e Timestamp. Il primo, Testo, rappresenta una label testuale che verrà visualizzata dall'utente per ricondurre al discorso di riferimento nella chat. Il secondo ed ultimo attributo, Timestamp, a sua volta generato dal server indica l'orario di risposta in formato numerico con precisione al secondo.

Il legame che esprime il destinatario del messaggio è reso esplicito dalla relazione “riceve” di tipo uno a molti dato che un messaggio è indirizzato ad uno ed un solo dipendente e viceversa un dipendente può aver ricevuto molti messaggi come nessuno - *nel caso in cui non abbia ancora usufruito di tale servizio* -. La chiave di *Messaggio* è una chiave composta dalla chiave esterna di dipendente (CF) e dal Timestamp per rispettare il vincolo che il messaggio è riferito ad un dipendente.

Schema ER



Progettazione Logica

Dal momento in cui non esistono DBMS in grado di operare direttamente sugli oggetti di uno schema E/R, è necessario tradurre lo schema in modelli di dati supportati da DBMS

La progettazione logica procede secondo il seguente ordine:

1. Eliminazione gerarchie ISA
2. Selezione chiavi primarie, eliminazione identificazioni esterne
3. Trasformazione attributi composti o multipli
4. Traduzione di entità e associazioni in schemi e relazioni
5. Verifica di normalizzazione.

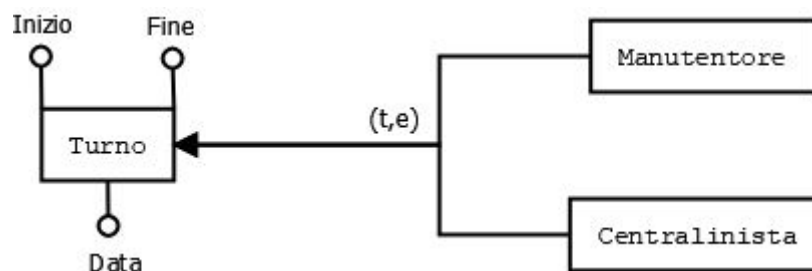
Eliminazione gerarchie ISA

Per quanto riguarda l'eliminazione delle gerarchie, si valutano essenzialmente 3 metodologie:

1. mantenimento delle entità con associazioni
2. collasso verso l'alto
3. collasso verso il basso

Turno

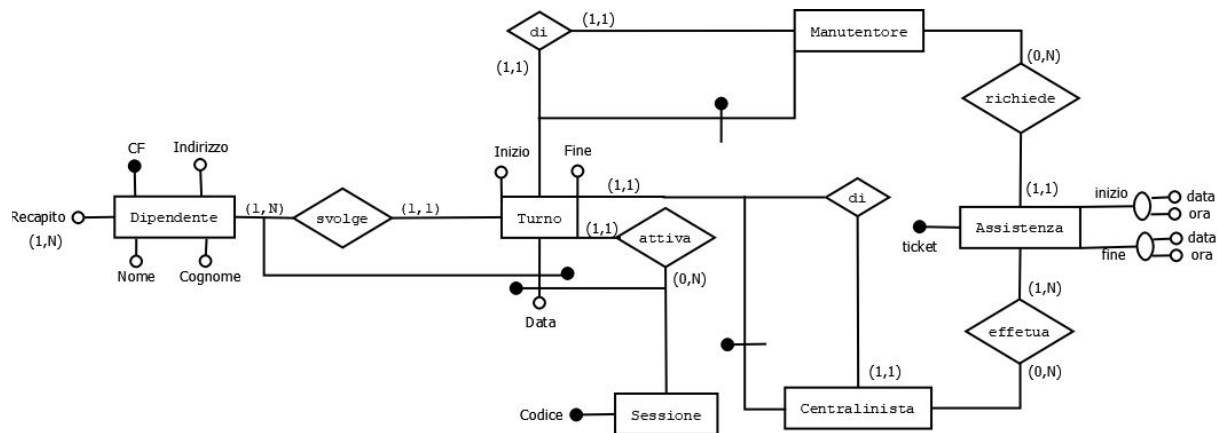
La gerarchia di interesse riguarda la generalizzazione effettuata tramite l'entità Turno, che si specializza in Manutentore e Centralinista. La scelta di Turno anziché Dipendente è spiegata nella progettazione concettuale.



Dal momento in cui non ci sono molti attributi specializzati si potrebbe pensare ad un collasso verso l'alto, ma ciò non è la scelta ottimale in quanto complicherebbe la gestione delle relazioni tra Centralinista e Assistenza oppure tra Manutentore e Inventario (o Mansione). In aggiunta ciò introdurrebbe anche una relazione auto-ricorsiva per via della relazione delle entrambe con Assistenza. Inoltre non è una soluzione lungimirante, poiché complicherebbe future modifiche al database specifiche solo a un Centralinista oppure Manutentore e relative relazioni. Essendo la gerarchia totale ed esclusiva si può pensare ad un collasso verso il basso, a discapito però della ridondanza e in aggiunta c'è da considerare che l'entità Turno è relazionata con Dipendente che a sua volta è relazionata con Messaggi. Tuttavia, in presenza di molti attributi di specializzazione questa metodologia favorisce le operazioni in cui si accede separatamente a ciascuna delle entità

figlie. Infine, si può sempre e comunque seguire la metodologia del mantenimento delle entità. Si è scelta proprio quest'ultima soluzione, in quanto ci sono relazioni associate a Turno.

Complessivamente lo schema risulta nel seguente modo

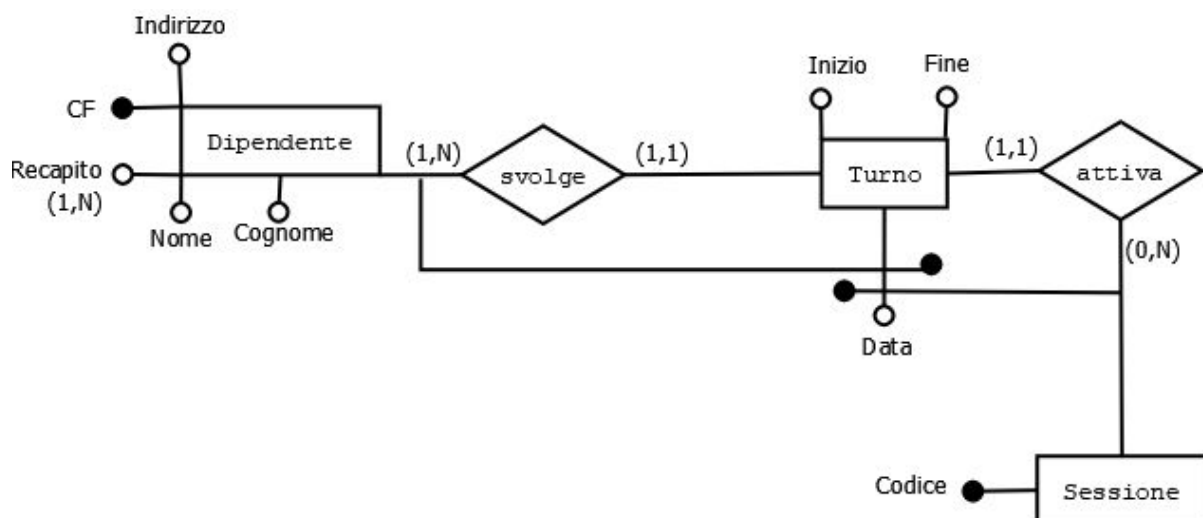


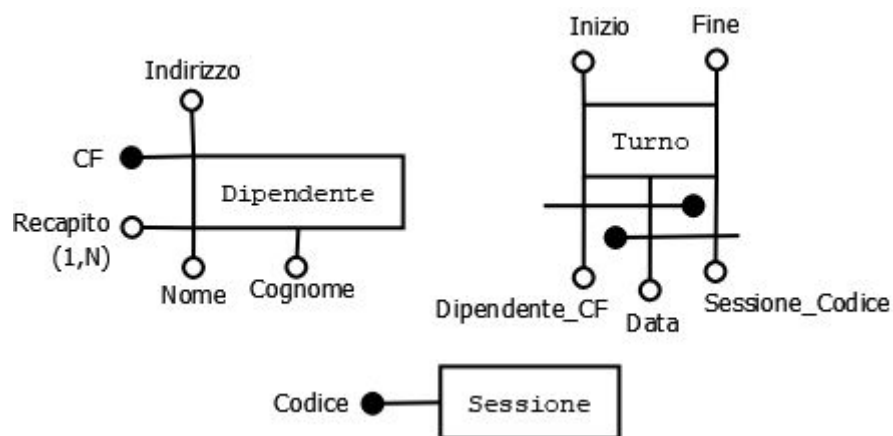
Selezione delle chiavi primarie ed eliminazione delle chiavi esterne

È necessario eliminare le identificazione esterne dato che per definizione si ha che: una componente di identificazione esterna di una entità E2 da una entità E1 tramite una associazione R comporta il trasporto dell'identificatore di E1 su E2.

Dipendente - Turno

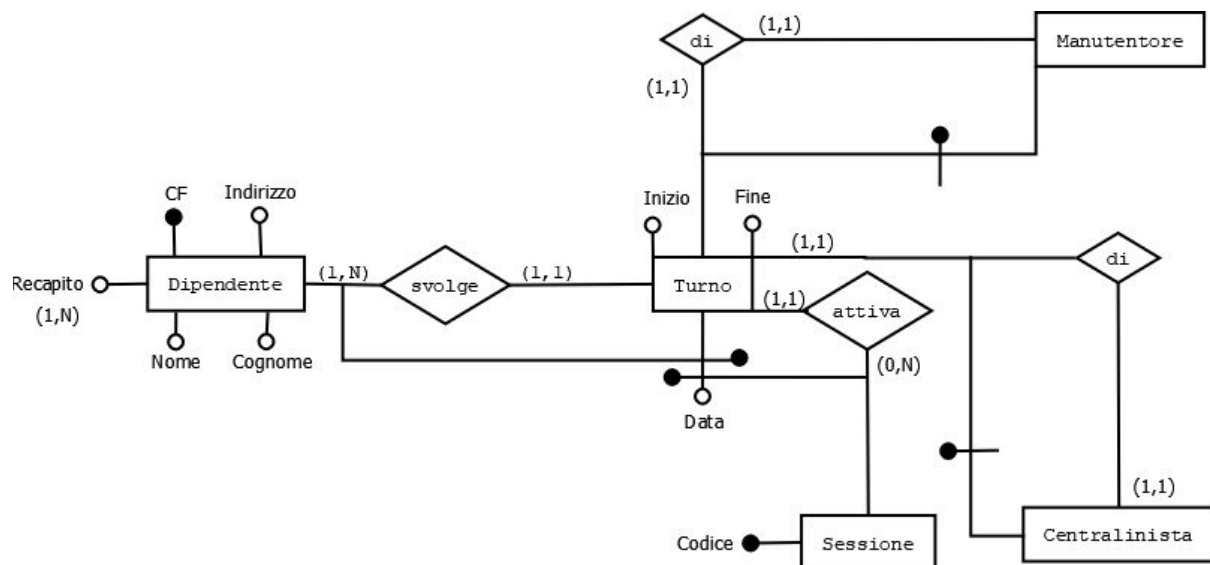
La entità Turno ha una componente di identificazione esterna dall'entità Dipendente e una dall'entità Sessione. È necessario effettuare la seguente trasformazione

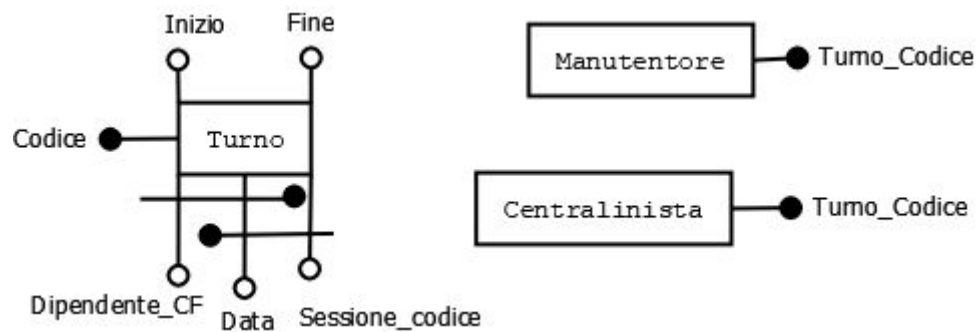




Centralinista e Manutentore

Le entità Centralinista e Manutentore hanno componenti di identificazione dall'entità Turno. Dunque è necessario effettuare la seguente trasformazione (si prendono in considerazione solo le entità Centralinista e Manutentore in quanto le altre sono state già precedentemente trattate).

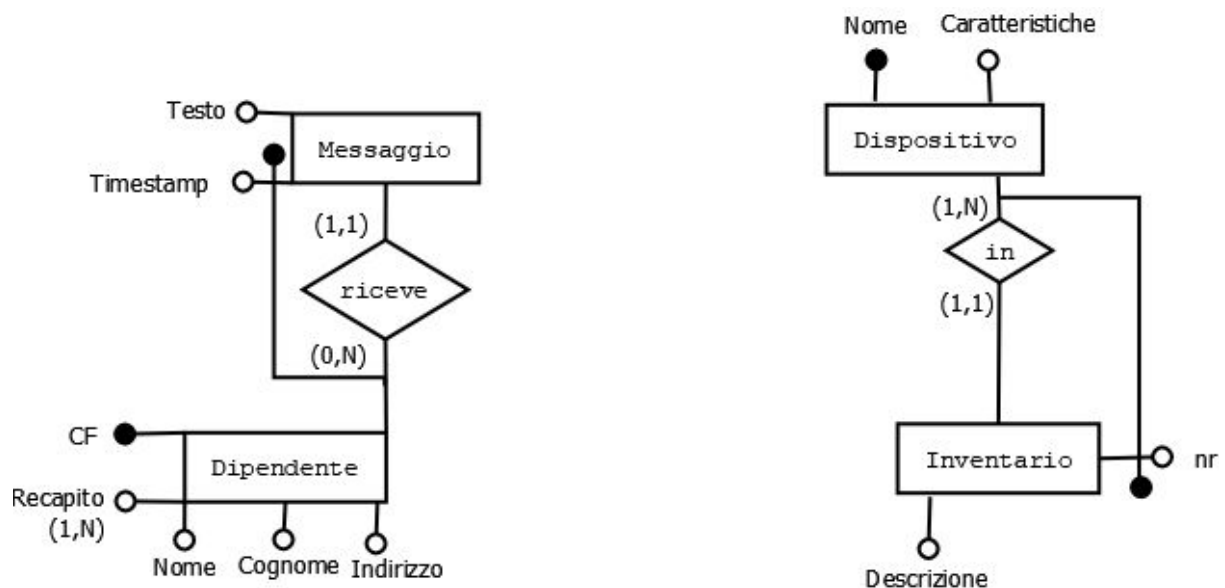


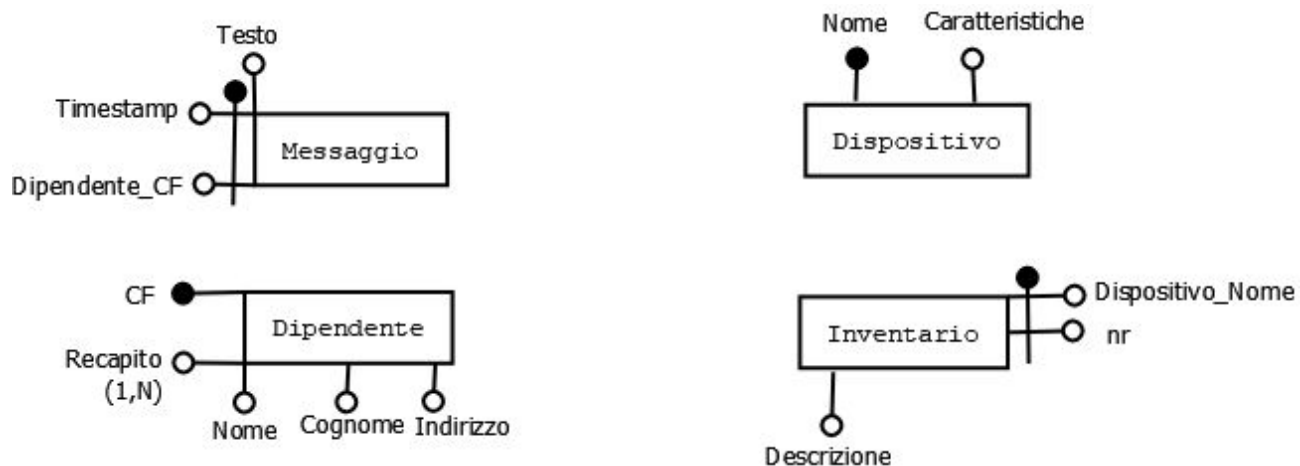


Si è pensato di aggiungere una chiave alternativa Codice, poiché semplifica notevolmente la gestione delle chiavi in particolare sulle entità figlie (sopra raffigurate) ma anche su tutte le altre entità in relazione con esse. Naturalmente, non riportando la data, il vincolo che esprimere l'appartenenza di una data attività - *quale essa sia un prestito, un'assistenza o un intervento* - al turno in questione dovrà essere gestito tramite ulteriori controlli quali ad esempio trigger. Questo approccio infine migliora la correttezza dei dati evitando di fare confusione tra la data del turno e la data delle attività.

Messaggio e Inventario

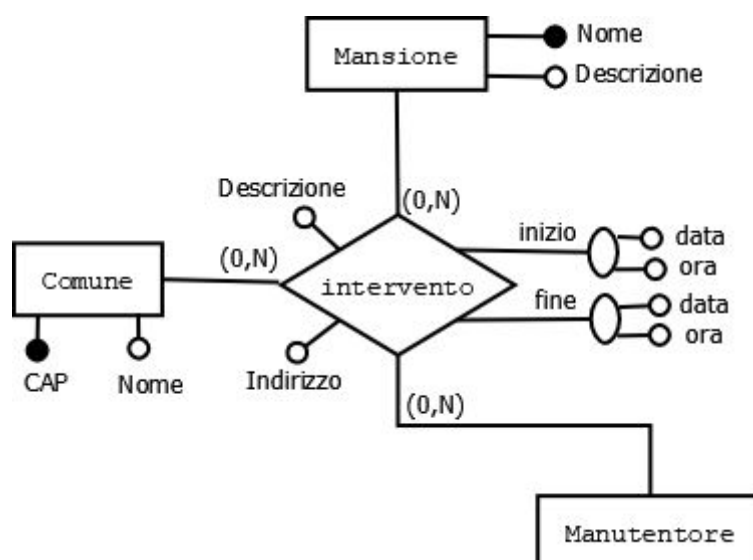
L'entità Messaggio ha una componente di identificazione esterna con Dipendente. Inoltre l'entità Inventario ha una identificazione esterna con Dispositivo. È necessario effettuare la seguenti modifiche.

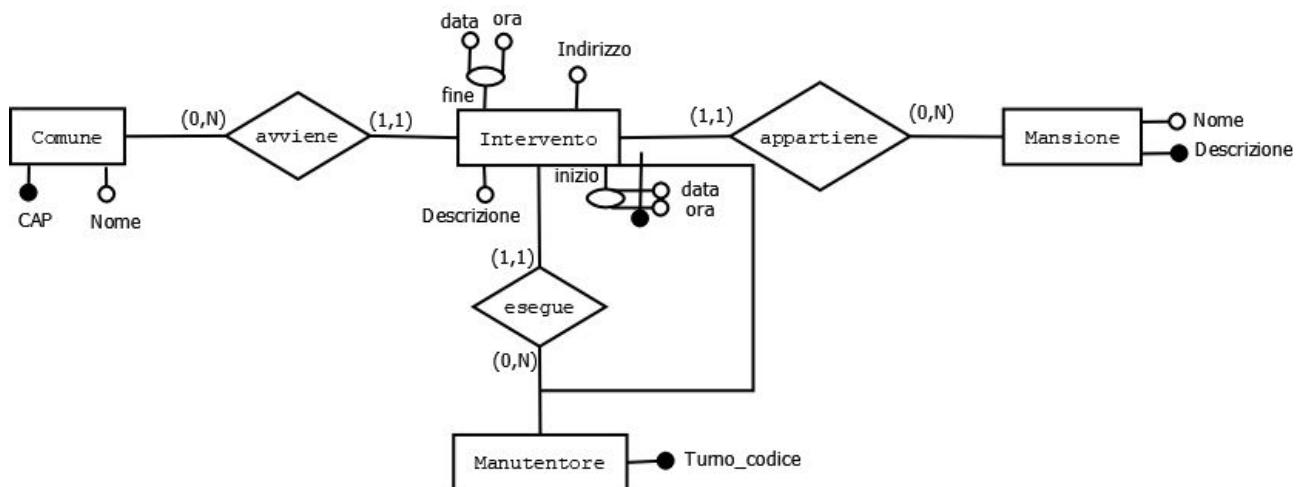




Intervento

È necessaria una reificazione e un raffinamento per ciò che riguarda l'associazione, come spiegato nella progettazione concettuale. In particolare si ha la seguente struttura di partenza





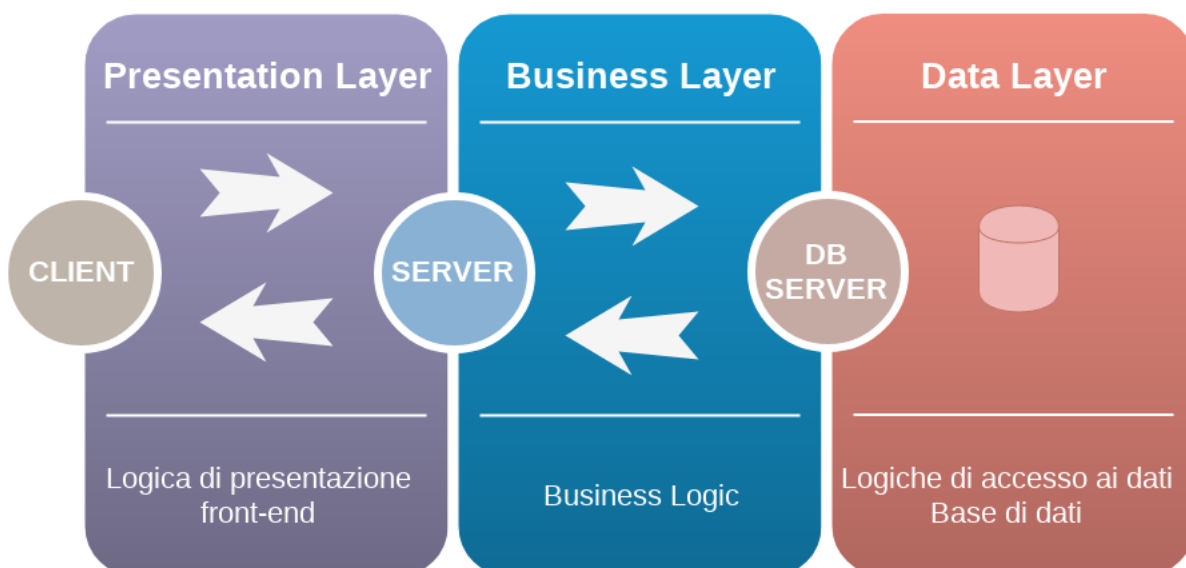
L'entità Intervento deve sottostare ai seguenti vincoli:

1. Un manutentore non può fare altri interventi (indipendentemente dalla mansione) una volta che ha iniziato un determinato intervento.
2. Non ci possono essere più interventi dello stesso manutentore sovrapposti a livello temporale.

Il secondo vincolo è una conseguenza del primo, tuttavia per entrambi i casi bisogna fare un discorso più complesso trattato nel paragrafo successivo.

Vincolo con trigger

Si può prevedere un trigger passivo per far rispettare il primo vincolo. In particolare ciò che bisognerà fare è che ad ogni inserimento si vada a guardare tutti gli interventi per quel manutentore che hanno la data fine e ora fine null. Se il risultato è vuoto allora l'inserimento viene effettuato in caso contrario no. Sarà proprio questo l'approccio utilizzato con l'obiettivo finale di semplificare la Business Logic scaricando questi controlli all'interno del Data Layer gestito dal database admin. Per rendere più chiaro il concetto, si consideri il seguente modello descrittivo di una comune applicazione web



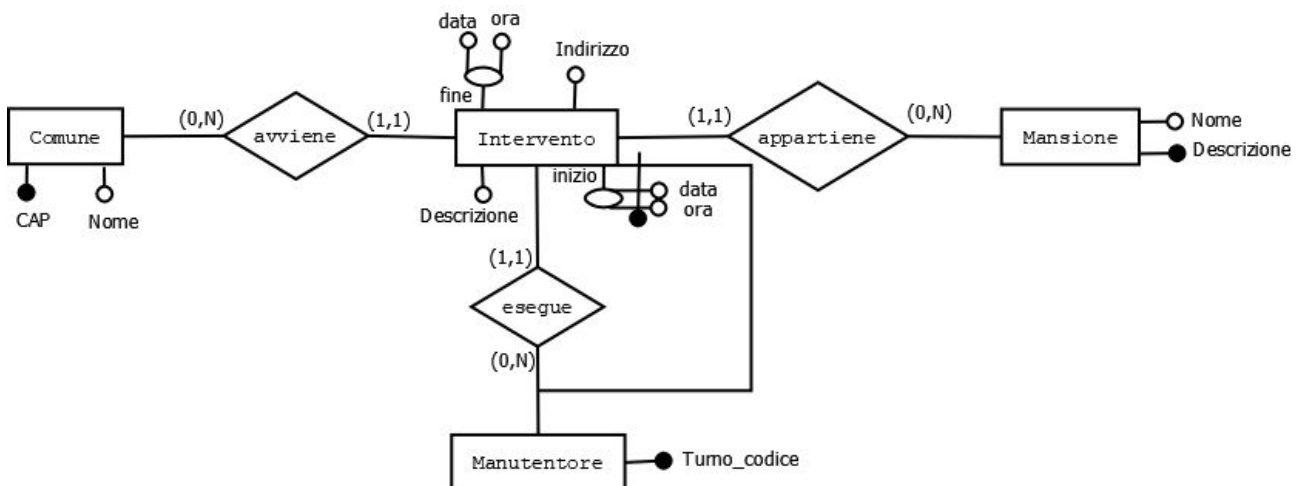
Spostando alcuni concetti del Business Layer nel Data Layer (tramite stored procedure, triggers) si ha un aumento di responsabilità dell'amministratore della base di dati rispetto allo sviluppatore. Generalizzando il concetto, nel caso in cui sia frequente questa scelta per ogni parte della propria business logic ci possono essere degli svantaggi. I principali:

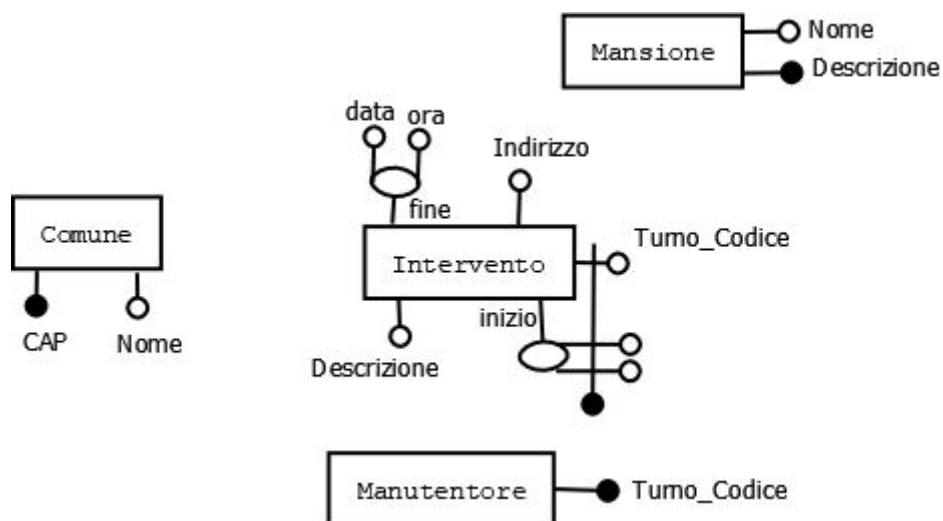
- Scalabilità: il Business Layer è molto più flessibile e molto più facilmente scalabile rispetto al Data layer.
- Manutenibilità: avere un codice leggibile e facilmente comprensibile cosa che non è del tutto immediata per quanto riguarda il Data layer tra triggers e stored procedure che non hanno la stessa espressività rispetto a linguaggi usati nel Business Layer (basti pensare a PHP). Ciò si ripercuote anche nell'azienda stessa che dovrà ricercare (con costi maggiori) figure DB Admin altamente specializzate.

Tuttavia, ci sono anche diversi vantaggi:

- Avere una Business logic centralizzata nel caso si usino stored procedure per realizzarla porta alla diretta conseguenza di essere facilmente usabile da qualsiasi applicativo.
- Sfruttare appieno le funzionalità di un RDBMS, fatto apposta per la gestione e manipolazione dei dati.
- Aumento di riutilizzo, efficienza e controllo (sui dati).

Fatte le considerazioni sopra illustrate, si procede con l'eliminazione delle chiavi esterne. Infatti l'entità Intervento ha identificazione esterna con Manutentore. Si ha quindi la seguente trasformazione





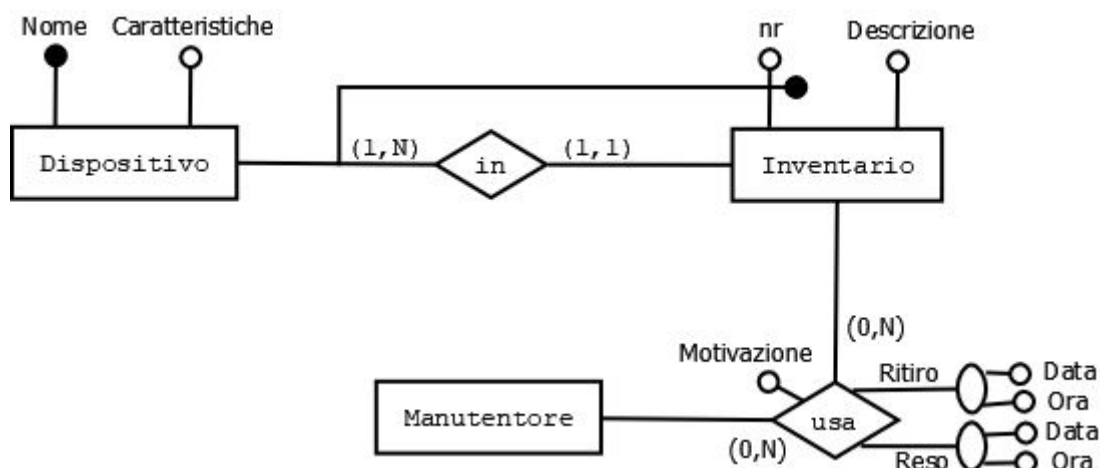
Si potrebbe pensare a una chiave alternativa Intervento_Codice per l'entità Intervento ma tuttavia si suppone che almeno la data e il codice turno del manutentore sia specificato nelle interrogazioni. La chiave alternativa Turno_Codice è già stata discussa precedentemente.

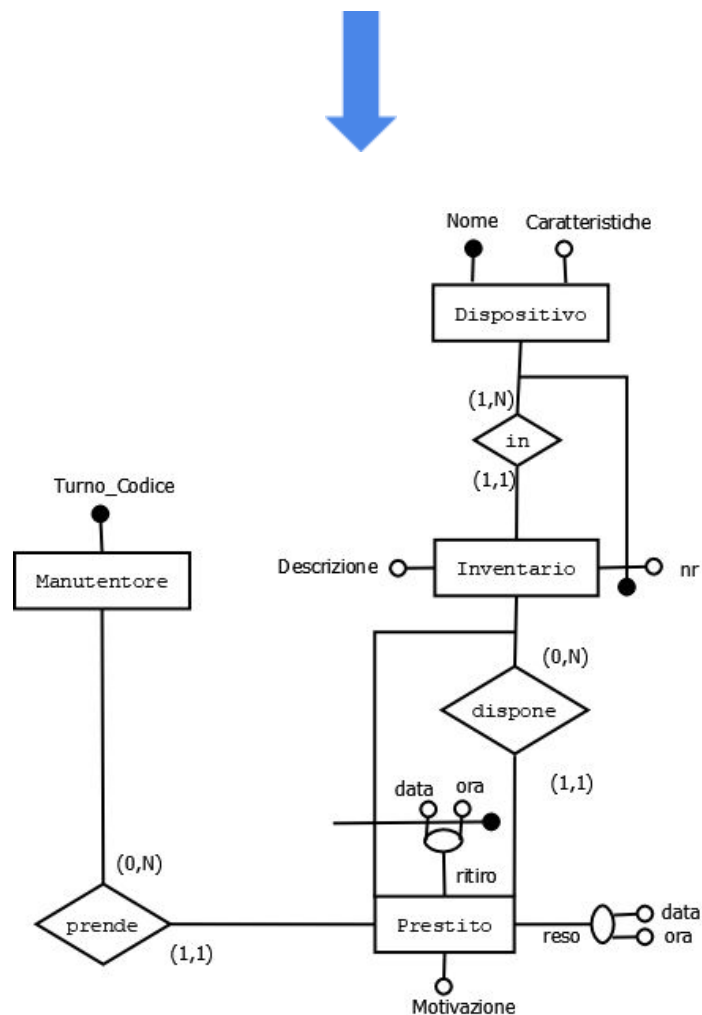
Prestito

È necessario effettuare una reificazione della relazione “prestito” poiché bisogna imporre determinati vincoli e chiavi. Si hanno dunque i seguenti vincoli:

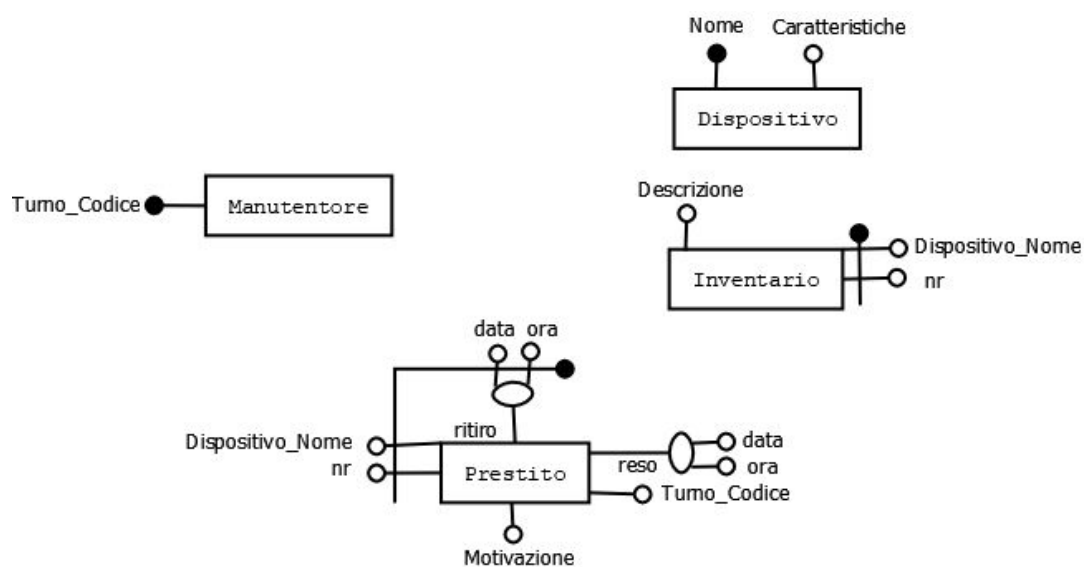
1. Un oggetto non può pretato nuovamente fino a quando quest'ultimo non è stato restituito.
2. Non ci possono essere più prestiti dello stesso oggetto sovrapposti a livello temporale.

Anche per questi vincoli, si rimanda alla discussione relativa all'Intervento dove si è discusso di vincoli simili risolti attraverso trigger. Per quanto riguarda la reificazione si ha la seguente trasformazione





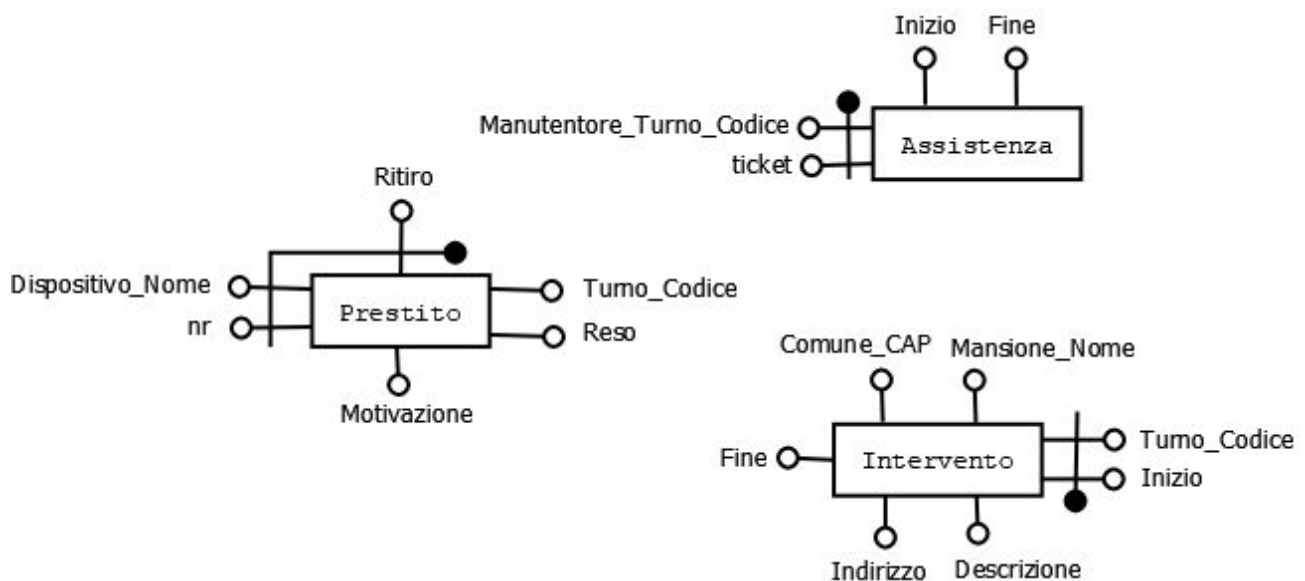
È necessario eliminare le identificazioni esterne, infatti l'entità **Prestito** ha parte dell'identificazione esterna da parte dell'entità **Manutentore** e dall'entità **Inventario**. Si attua quindi la seguente trasformazione



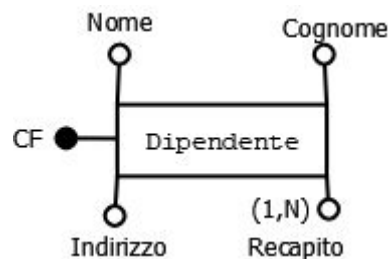
Da notare la chiave composta su Prestito. Per rispettare la proprietà di minimalità della chiave e allo stesso tempo per rispettare parte dei vincoli richiesti si è deciso di comporre la data e ora di ritiro con l'identificazione esterna di dispositivo ed escludere la chiave del turno di riferimento del dipendente in questione.

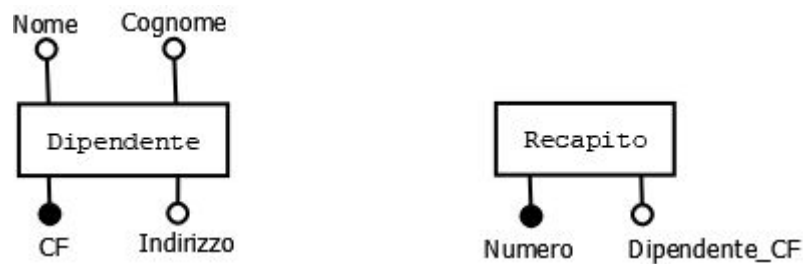
Trasformazione attributi composti o multipli

Sono stati risolti gli attributi composti - *data e ora* - delle entità Intervento (inizio e fine), Prestito (ritiro e reso) e Assistenza (inizio e fine). Si è scelto di collassare l'attributo composto in uno singolo al fine di semplificare e rendere più immediata la gestione di questi valori all'interno del database appoggiandosi al tipo di dato *timestamp*. Sono quindi state effettuate le seguenti trasformazioni



È stato inoltre risolto l'attributo multiplo Recapito dall'entità Dipendente



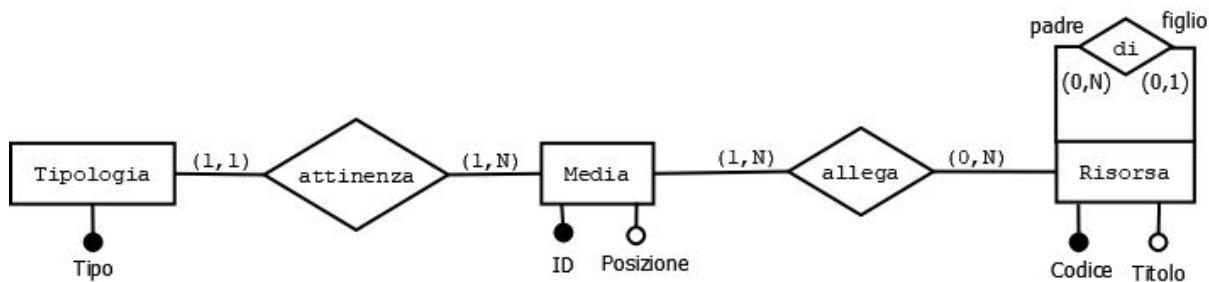


Infine si è dovuto risolvere l'attributo composto Pericolo dell'entità Gruppo



Traduzione di entità e associazioni in schemi e relazioni

Si vuole tradurre le entità e relazioni in schemi relazionali. Nelle traduzioni, se si incontrano entità non tradotte nello schema sottostante significherà che sono già state tratte in precedenza.



Siccome le entità Tipologia e Media sono legate da una associazione 1:N è possibile accorpare l'associazione all'entità Media.

Si ha una associazione N:M tra l'entità Media e Risorsa, quindi sono state tradotte con una relazione per ogni entità e una per l'associazione

Infine, l'entità Risorsa ha una relazione auto-ricorsiva 1:N, traducibile accorpendo dalla parte del padre

Si ha quindi il seguente schema

TIPOLOGIA (Tipo)

MEDIA (ID, Posizione, Tipologia_Tipo)

FK: Tipologia_Tipo REFERENCES TIPOLOGIA

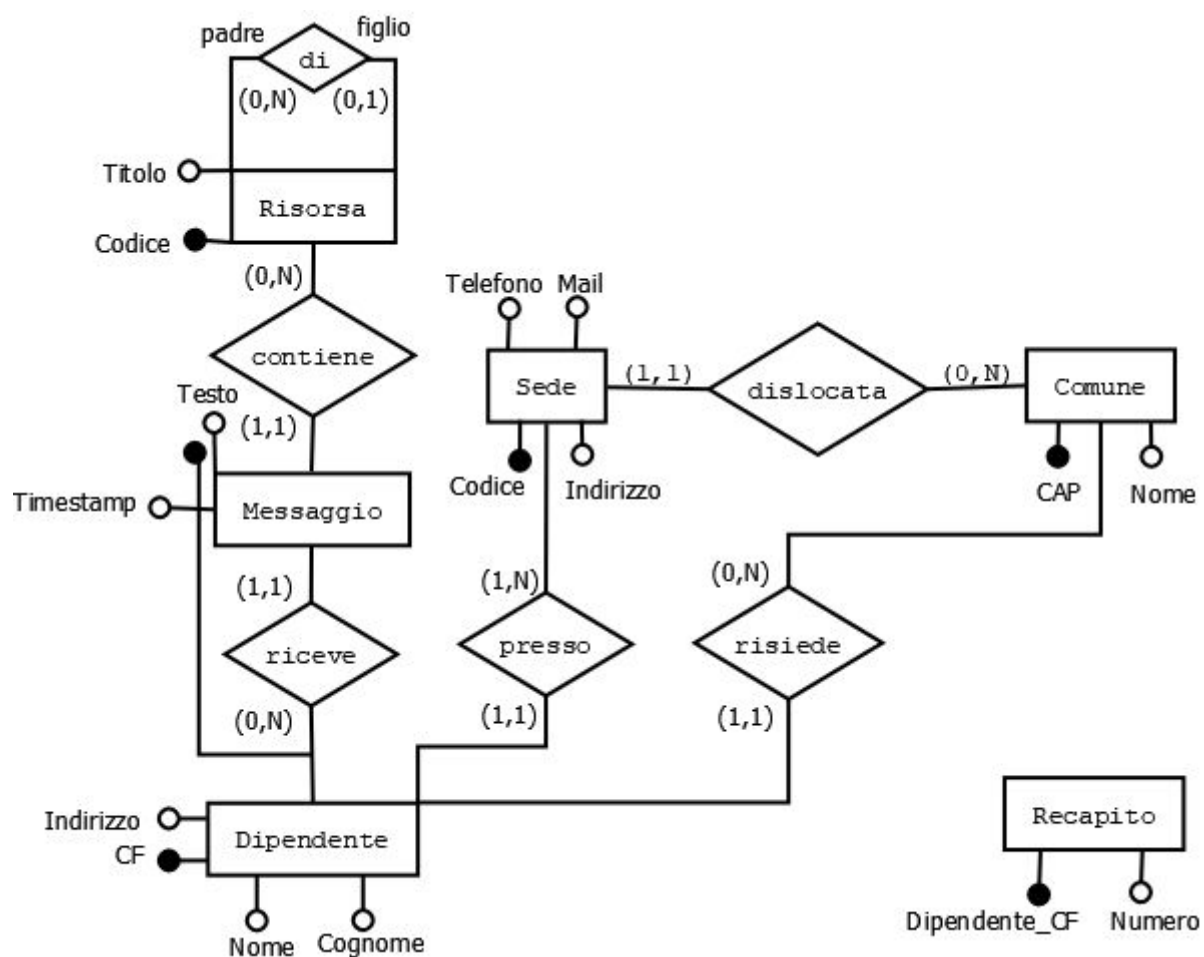
RISORSA (Codice, Titolo, Padre)

FK: Padre REFERENCES RISORSA

ALLEGATO (Risorsa_Codice, Media_ID)

FK: Risorsa_Codice REFERENCES RISORSA

FK: Media_ID REFERENCES MEDIA



Le entità Sede e Comune sono associate da una relazione 1:N, si accorpa l'associazione in Sede

Le entità Dipendente e Comune sono associate da una relazione 1:N, si accorpa l'associazione in Sede

Le entità Dipendente e Sede sono legate da una associazione 1:N, quindi si accorpa l'associazione "presso" in Dipendente

La entità Recapito viene trasformata con la traduzione standard

Le entità Dipendente e Messaggio sono legate da una associazione 1:N, quindi si accorpa l'associazione in Messaggio

Le entità Messaggio e Risorsa sono collegate tramite una associazione 1:N, quindi si accorpa l'associazione in Messaggio

COMUNE (CAP, Nome)

SEDE (Codice, Indirizzo, Mail, Telefono, Comune_CAP)

FK: Comune_CAP REFERENCES COMUNE

DIPENDENTE (CF, Nome, Cognome, Indirizzo, Comune_CAP, Sede_Codice)

FK: Comune_CAP REFERENCES COMUNE

FK: Sede_Codice REFERENCES SEDE

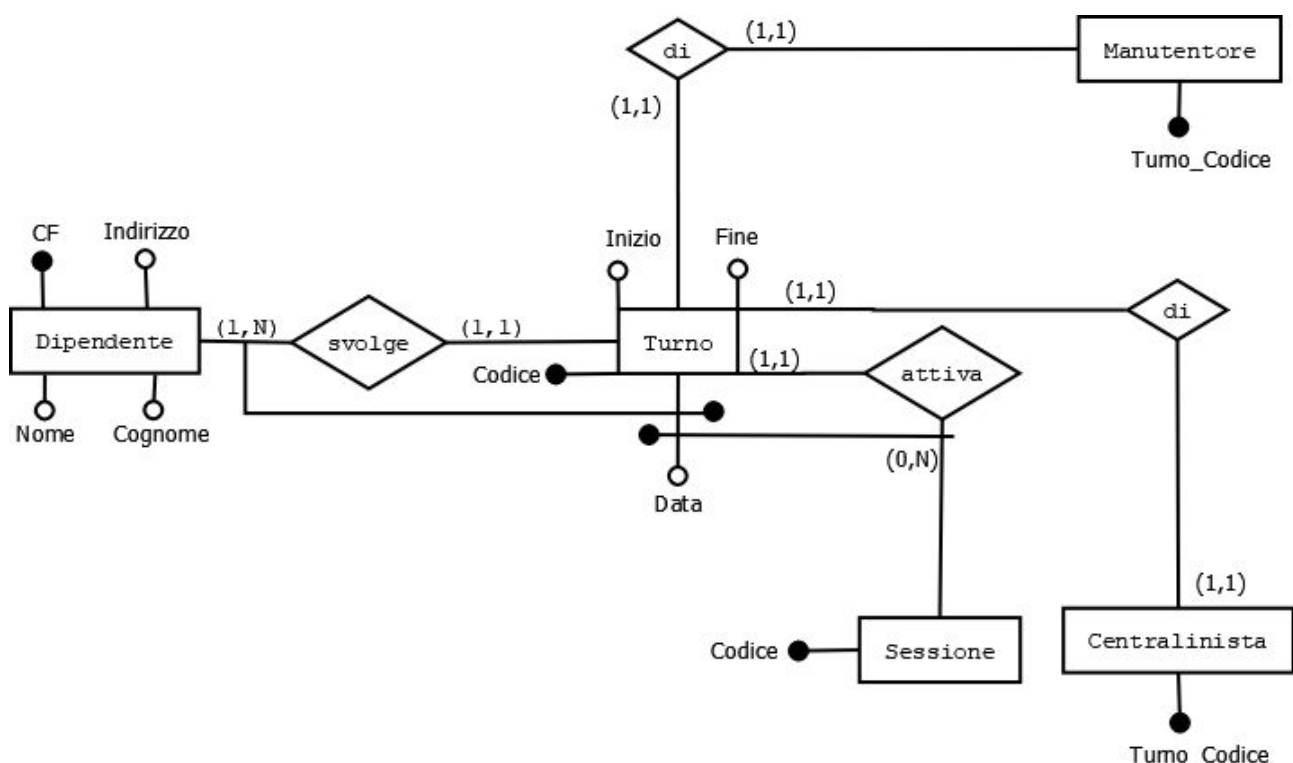
RECAPITO (Numero, Dipendente_CF)

FK: Dipendente_CF REFERENCES DIPENDENTE

MESSAGGIO (Timestamp, Testo, Risorsa_Codice, Dipendente_CF)

FK: Risorsa_Codice REFERENCES RISORSA

FK: Dipendente_CF REFERENCES DIPENDENTE



Le entità Dipendente e Turno sono associate da una relazione 1:N, si accorpa l'associazione in Turno

Le entità Turno e Sessione sono associate da una relazione 1:N, si accorpa l'associazione in Turno

Le entità Manutentore e Turno sono legate da una associazione 1:1, si procede con la traduzione standard con l'accorpamento su Manutentore

Le entità Centralinista e Turno sono legate da una associazione 1:1, si procede con la traduzione standard con l'accorpamento su Centralinista.

SESSIONE (Codice)

TURNO (Codice, Dipendente_CF, Sessione_Codice, Data, Inizio, Fine)

FK: Dipendente_CF REFERENCES DIPENDENTE

FK: Sessione_Codice REFERENCES SESSIONE

AK: Data, Dipendente_CF

AK: Data, Session_Codice

MANUTENTORE (Turno_Codice)

FK: Turno_Codice REFERENCES TURNO

CENTRALINISTA (Turno_Codice)

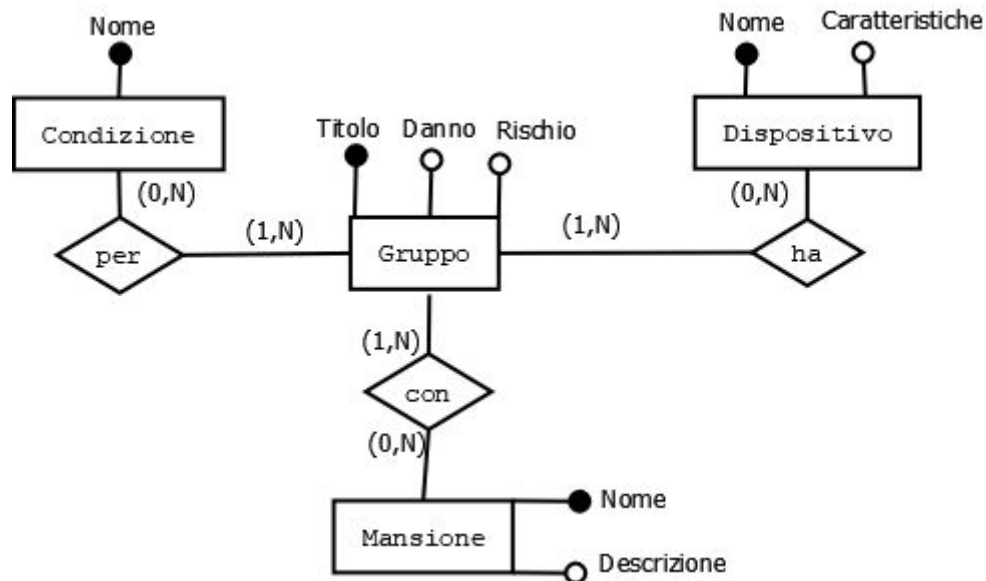
FK: Turno_Codice REFERENCES TURNO



L'associazione "describe" è di tipo 1:1 opzionale. Trattandosi di una associazione binaria parziale è si è preferito accorpare l'entità Risorsa sotto Mansione, in quanto sono previsti casi di Mansione che non abbiano una risorsa.

MANSIONE (Nome, Descrizione, Risorsa_Codice)

FK: Risorsa_Codice REFERENCES RISORSA



Tutte le entità presenti: Condizione e Gruppo, Dispositivo e Gruppo, Mansione e Gruppo sono in relazione tra loro con una associazione N:M. In tutti i casi si procede con la traduzione standard delle associazioni

CONDIZIONE (Nome)

DISPOSITIVO (Nome, Caratteristiche)

GRUPPO (Titolo, Danno, Rischio)

CONDIZIONE_GRUPPO (Condizione_Nome, Gruppo_Titolo)

FK: Condizione_Nome REFERENCES CONDIZIONE

FK: Gruppo_Titolo REFERENCES GRUPPO

DISPOSITIVO_GRUPPO (Dispositivo_Nome, Gruppo_Titolo)

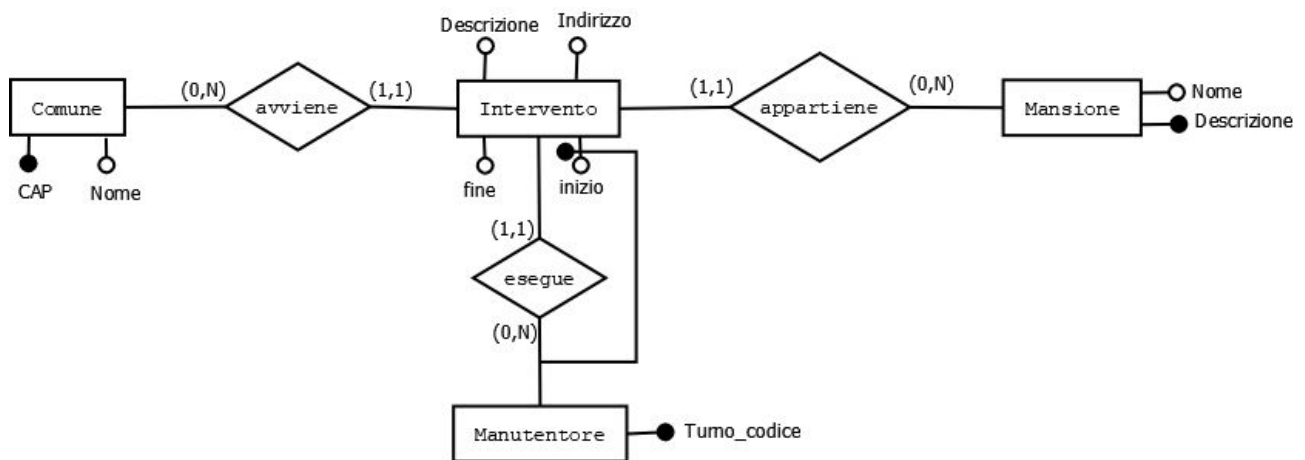
FK: Dispositivo_Nome REFERENCES DISPOSITIVO

FK: Gruppo_Titolo REFERENCES GRUPPO

MANSIONE_GRUPPO (Mansione_Nome, Gruppo_Titolo)

FK: Mansione_Nome REFERENCES MANSIONE

FK: Gruppo_Titolo REFERENCES GRUPPO



L'associazione "avviene" è di tipo 1:N, si effettua una traduzione standard delle entità legate e si accorpa l'associazione all'entità Intervento

L'associazione "appartiene" è di tipo 1:N, si effettua una traduzione standard delle entità legate e si accorpa l'associazione all'entità Intervento

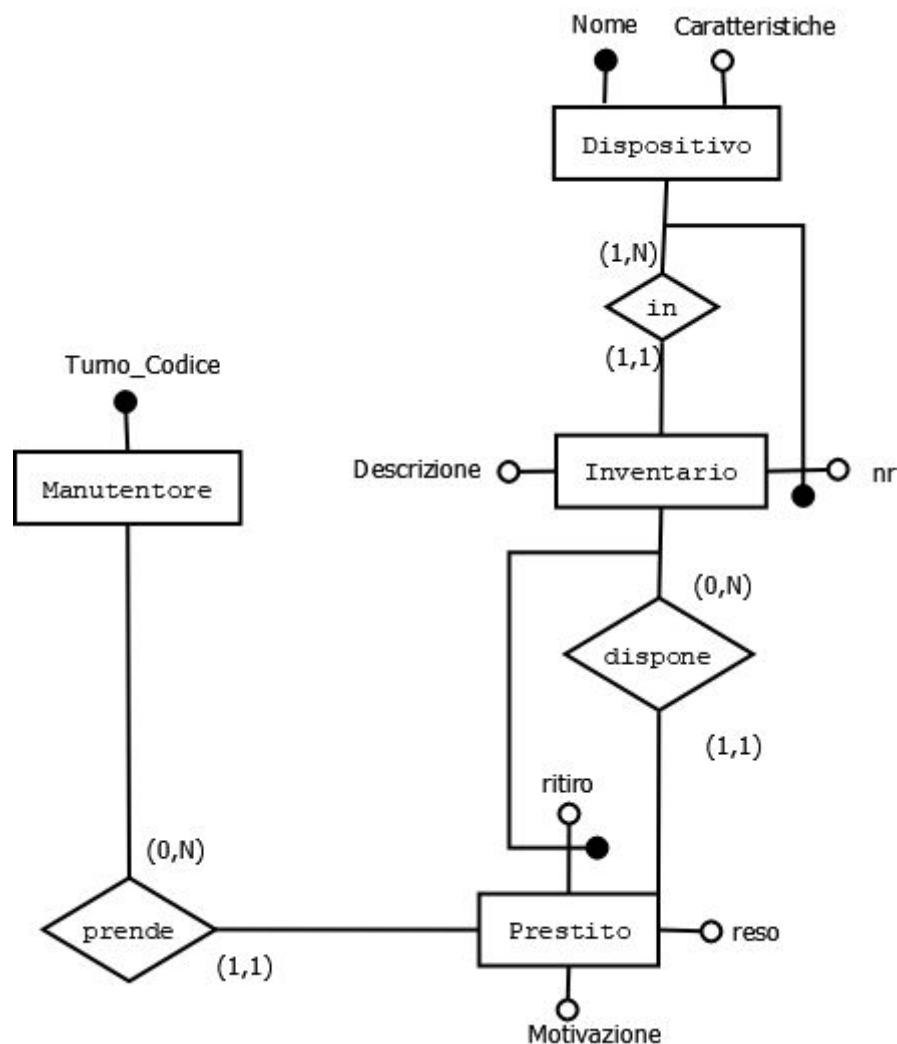
L'associazione "esegue" è di tipo 1:N, si effettua una traduzione standard delle entità legate e si accorpa l'associazione all'entità Intervento

INTERVENTO (Manutentore_Turno_Codice, Mansione_Nome, Comune_CAP, Indirizzo, Descrizione, Inizio, Fine)

FK: Manutentore_Turno_Codice REFERENCES MANUTENTORE

FK: Mansione_Nome REFERENCES MANSIONE

FK: Comune_CAP REFERENCES COMUNE



Le entità Dispositivo e Inventario sono associate tramite una relazione 1:N, si procede con la traduzione standard delle entità e l'accorpamento dell'associazione nell'entità Inventario

L'associazione "dispone" è di tipo 1:N, si procede con la traduzione standard delle entità e l'accorpamento dell'associazione nell'entità Prestito

L'associazione "prende" è di tipo 1:N, si procede con la traduzione standard delle entità e l'accorpamento dell'associazione nell'entità Prestito

INVENTARIO (Nr, Dispositivo_Nome, Descrizione)

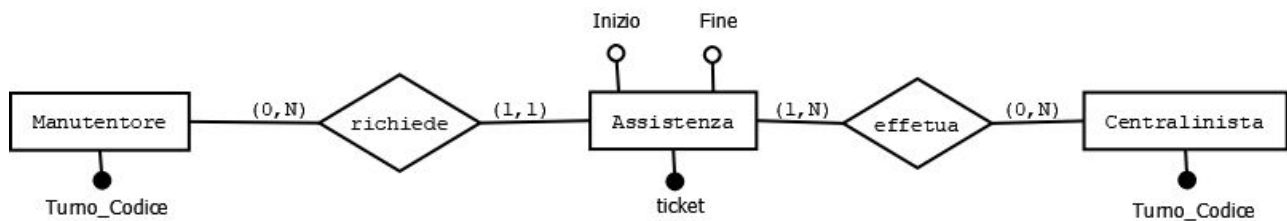
FK: Dispositivo_Nome REFERENCES DISPOSITIVO

PRESTITO (Manutentore_Turno_Codice, Inventario_Nr, Inventario_Dispositivo_Nome, Motivazione, Ritiro, Reso)

FK: Manutentore_Turno_Codice REFERENCES MANUTENTORE

FK: Inventario_Nr REFERENCES INVENTARIO

FK: Inventario_Dispositivo_Nome REFERENCES INVENTARIO



L'associazione "richiede" è di tipo 1:N, si procede con la traduzione standard delle entità e l'accorpamento dell'associazione all'entità Assistenza

L'associazione "effetua" è di tipo N:M, si procede con la traduzione standard delle entità.

ASSISTENZA (Manutentore_Turno_Codice, Ticket, Inizio, Fine)

FK: Manutentore_Turno_Codice REFERENCES MANUTENTORE

PARTECIPANTE (Assistenza_Ticket, Centralinista_Turno_Codice)

FK: Assistenza_Ticket REFERENCES ASSISTENZA

FK: Centralinista_Turno_Codice REFERENCES CENTRALINISTA

Schema logico completo

TIPOLOGIA (Tipo)

MEDIA (ID, Posizione, Tipologia_Tipo)

FK: Tipologia_Tipo REFERENCES TIPOLOGIA

RISORSA (Codice, Titolo, Padre)

FK: Padre REFERENCES RISORSA

ALLEGATO (Risorsa_Codice, Media_ID)

FK: Risorsa_Codice REFERENCES RISORSA

FK: Media_ID REFERENCES MEDIA

COMUNE (CAP, Nome)

SEDE (Codice, Indirizzo, Mail, Telefono, Comune_CAP)

FK: Comune_CAP REFERENCES COMUNE

DIPENDENTE (CF, Nome, Cognome, Indirizzo, Comune_CAP, Sede_Codice)

FK: Comune_CAP REFERENCES COMUNE

FK: Sede_Codice REFERENCES SEDE

RECAPITO (Numero, Dipendente_CF)

FK: Dipendente_CF REFERENCES DIPENDENTE

MESSAGGIO (Timestamp, Testo, Risorsa_Codice, Dipendente_CF)

FK: Risorsa_Codice REFERENCES RISORSA

FK: Dipendente_CF REFERENCES DIPENDENTE

SESSIONE (Codice)

TURNO (Codice, Dipendente_CF, Sessione_Codice, Data, Inizio, Fine)

FK: Dipendente_CF REFERENCES DIPENDENTE

FK: Sessione_Codice REFERENCES SESSIONE

AK: Data, Dipendente_CF

AK: Data, Session_Codice

MANUTENTORE (Turno_Codice)

FK: Turno_Codice REFERENCES TURNO

CENTRALINISTA (Turno_Codice)

FK: Turno_Codice REFERENCES TURNO

CONDIZIONE (Nome)

DISPOSITIVO (Nome, Caratteristiche)

MANSIONE (Nome, Descrizione, Risorsa_Codice)

FK: Risorsa_Codice REFERENCES RISORSA

GRUPPO (Titolo, Danno, Rischio)

CONDIZIONE_GRUPPO (Condizione_Nome, Gruppo_Titolo)

FK: Condizione_Nome REFERENCES CONDIZIONE

FK: Gruppo_Titolo REFERENCES GRUPPO

DISPOSITIVO_GRUPPO (Dispositivo_Nome, Gruppo_Titolo)

FK: Dispositivo_Nome REFERENCES DISPOSITIVO

FK: Gruppo_Titolo REFERENCES GRUPPO

MANSIONE_GRUPPO (Mansione_Nome, Gruppo_Titolo)

FK: Mansione_Nome REFERENCES MANSIONE

FK: Gruppo_Titolo REFERENCES GRUPPO

INTERVENTO (Manutentore_Turno_Codice, Mansione_Nome, Comune_CAP, Indirizzo, Descrizione, Inizio, Fine)

FK: Manutentore_Turno_Codice REFERENCES MANUTENTORE

FK: Mansione_Nome REFERENCES MANSIONE

FK: Comune_CAP REFERENCES COMUNE

INVENTARIO (Nr, Dispositivo_Nome, Descrizione)

FK: Dispositivo_Nome REFERENCES DISPOSITIVO

PRESTITO (Manutentore_Turno_Codice, Inventario_Nr, Inventario_Dispositivo_Nome, Motivazione, Ritiro, Reso)

FK: Manutentore_Turno_Codice REFERENCES MANUTENTORE

FK: Inventario_Nr REFERENCES INVENTARIO

FK: Inventario_Dispositivo_Nome REFERENCES INVENTARIO

ASSISTENZA (Manutentore_Turno_Codice, Ticket, Inizio, Fine)

FK: Manutentore_Turno_Codice REFERENCES MANUTENTORE

PARTECIPANTE (Assistenza_Ticket, Centralinista_Turno_Codice)

FK: Assistenza_Ticket REFERENCES ASSISTENZA

FK: Centralinista_Turno_Codice REFERENCES CENTRALINISTA

Verifica della normalizzazione

Dopo le opportune verifiche il database risulta in forma normale, quindi non ci sono cambiamenti da segnalare.

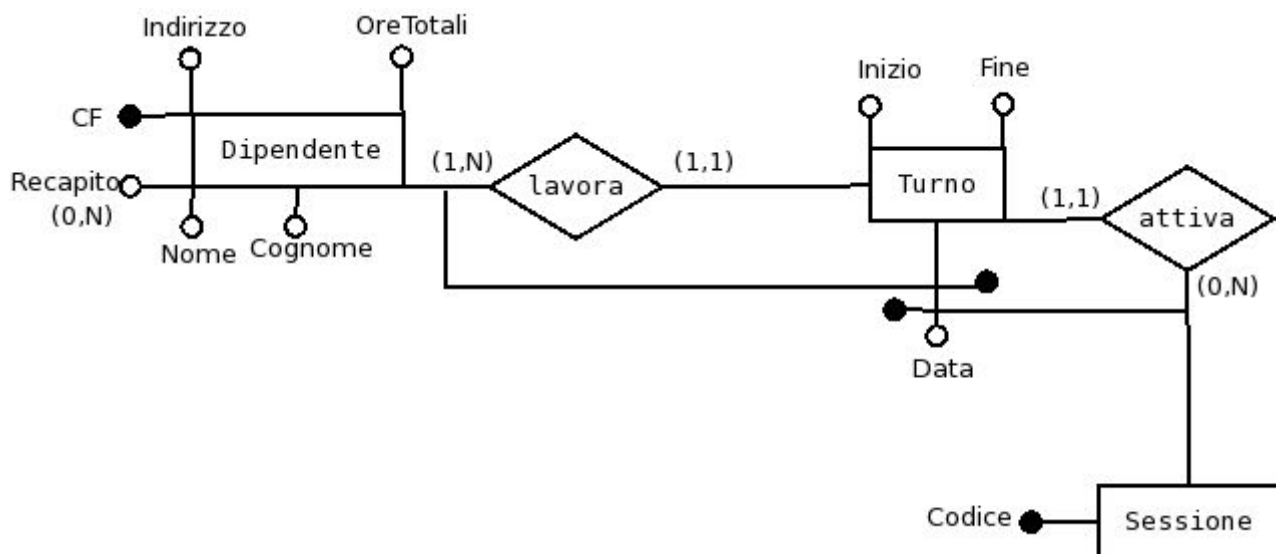
Vincoli aggiuntivi

Con la notazione “lasso di tempo” si intende una data di inizio e di fine nel formato timestamp (date e ora). Di seguito sono riportati i vincoli aggiuntivi che dovranno essere gestiti.

1. Verificare che all’inserimento di un media, quest’ultimo faccia riferimento solamente a risorse di tipo “foglia” nel contesto dell’albero
2. Verificare che all’inserimento di una task con documentazione, quest’ultima faccia riferimento solamente a risorse di tipo “foglia” nel contesto dell’albero
3. Verificare che venga rispettato il vincolo della generalizzazione esclusiva, ovvero che un turno sia esclusivamente di tipo centralinista o manutentore ma mai entrambe
4. Verificare che un manutentore o centralinista non partecipi ad assistenze al di fuori del suo orario lavorativo
5. Verificare che un manutentore non partecipi ad ulteriori assistenze se ne ha già una in corso. Questo vincolo esclude a priori che vi siano assistenze sovrapposte o in un lasso di tempo parzialmente coincidente
6. Verificare che un manutentore non possa effettuare interventi al di fuori del suo orario lavorativo
7. Verificare che un manutentore non effettui ulteriori interventi se ne ha già uno in corso. Questo vincolo esclude a priori che vi siano interventi sovrapposti o in un lasso di tempo parzialmente coincidente
8. Verificare che un prestito di dispositivo dal magazzino venga effettuato solamente durante il turno lavorativo del manutentore che lo richiede
9. Verificare che un dispositivo venga prestato ad ulteriori manutentori se quest’ultimo non è ancora stato restituito. Questo vincolo esclude a priori che vi siano prestiti sovrapposti o in un lasso di tempo parzialmente coincidente

Studio dati derivati

Sono possibili diversi dati derivati per quanto riguarda la progettazione della base di dati. Verrà fatto uno studio accurato con la relativa modifica al diagramma er solo con uno di questi. In particolare il dato riguarda il *numero totale di ore svolte da un dipendente*. Ad esempio, ciò può essere utile per quanto riguarda il reparto contabilità / analisi per realizzare metriche. Aggiungendo l'attributo OreTotali sull'entità Dipendente, lo schema modificato risulta nel seguente modo



Lo studio successivo valuterà se effettivamente conviene tenere questa modifica o meno.

Tabella delle operazioni

Si suppone che siano presenti circa 6000 dipendenti¹ e per semplicità si considera che tutti lavorino tra centralinisti e manutentori (no assenze/malattie). Infine si presuppone che almeno uno tra analista (sia esso una persona o una routine, per fare analisi di business) e il dipendente per (avere un rendiconto) visualizzi il numero totale di ore svolte al giorno.

Operazione	Tipo	Frequenza
Inserimento turno giornaliero di un dipendente	I	6000/gg
Visualizzazione ore svolte di un dipendente	I	2/gg

¹ 5786 dichiarati nel 2007 (wikipedia) - Autostrade per l'Italia

Carico di lavoro

Il numero di dipendenti, come detto precedentemente è 6000. Si considera un mese lavorativo composto di 20 giorni (per semplicità non si considerano ferie, festività). Si presume che ci siano già dati per un anno lavorativo per dipendente.

Concetto	Tipo	Volume dei dati
Dipendente	E	6000
Lavora	R	240
Turno	E	1440000

Tabella degli accessi con dato derivato

Si presume che si conosca già il codice del dipendente a cui si fa riferimento.

Costo totale	Operazione	Concetto	Accessi	Tipo
2	Visualizzazione ore svolte di un dipendente	Dipendente	1	L
6 * 6000 = 36000	Inserimento turno giornaliero di un dipendente	Turno	1	S
		Lavora	1	L
		Dipendente	1	L
			1	S

Tabella degli accessi senza il dato derivato

Costo totale	Operazione	Concetto	Accessi	Tipo
2*481=962	Visualizzazione ore svolte di un dipendente	Dipendente	1	L
		Lavoro	240	L
		Turno	240	L
2*6000 = 12000	Inserimento turno giornaliero di un dipendente	Turno	1	S

Con dato derivato si ha 36002 accessi al giorno, senza invece si hanno 12962 accessi al giorno.
Non conviene tenere il dato derivato

Base di dati

Per quanto riguarda lo schema e la struttura del database si adottano le seguenti regole:

- I nomi delle tabelle e tutti gli attributi sono tradotti in inglese
- Il nome di una tabella è nel seguente formato: *nometabella*, tutto minuscolo
- Il nome degli attributi possono avere soltanto lettere minuscole e _ per la separazione (ove necessario)
- Le foreign key sono nel seguente formato *entità_attributo*

Query di creazione

```
CREATE TABLE typology (  
  type VARCHAR(6) PRIMARY KEY  
);  
  
CREATE TABLE media (  
  id SERIAL PRIMARY KEY,  
  source VARCHAR(80) NOT NULL,  
  typology_type VARCHAR(6) NOT NULL,  
  FOREIGN KEY(typology_type) REFERENCES typology(type)  
  ON UPDATE CASCADE  
  ON DELETE NO ACTION  
);  
  
CREATE TABLE resource (  
  id SERIAL PRIMARY KEY,  
  title VARCHAR(100) NOT NULL,  
  parent INTEGER,  
  
  FOREIGN KEY(parent) REFERENCES resource(id)  
  ON UPDATE CASCADE  
  ON DELETE SET NULL  
);  
  
CREATE TABLE attachment (  
  resource_id INTEGER NOT NULL,  
  media_id INTEGER NOT NULL,  
  
  FOREIGN KEY(resource_id) REFERENCES resource(id)  
  ON UPDATE CASCADE
```

```
    ON DELETE CASCADE,
FOREIGN KEY(media_id) REFERENCES media(id)
    ON UPDATE CASCADE
    ON DELETE CASCADE,

PRIMARY KEY(resource_id, media_id)
);

CREATE TABLE town (
    cap CHAR(5) PRIMARY KEY,
    name VARCHAR(40) NOT NULL,

CHECK (cap SIMILAR TO '[0-9]{5}')
);

CREATE TABLE office (
    id SERIAL PRIMARY KEY,
    address VARCHAR(40) NOT NULL,
    mail VARCHAR(30) NOT NULL,
    phone VARCHAR(15) NOT NULL,
    town_cap CHAR(6) NOT NULL,

FOREIGN KEY(town_cap) REFERENCES town(cap)
    ON UPDATE CASCADE
    ON DELETE NO ACTION
);

CREATE TABLE employee (
    cf CHAR(16) PRIMARY KEY,
    name VARCHAR(30) NOT NULL,
    surname VARCHAR(30) NOT NULL,
    address VARCHAR(40) NOT NULL,
    town_cap CHAR(6) NOT NULL,
    office_id INTEGER NOT NULL,

FOREIGN KEY(town_cap) REFERENCES town(cap)
    ON UPDATE CASCADE
    ON DELETE NO ACTION,
```

```
FOREIGN KEY (office_id) REFERENCES office(id)
  ON UPDATE CASCADE
  ON DELETE SET NULL
);

CREATE TABLE contact (
  number VARCHAR(15) PRIMARY KEY,
  employee_cf CHAR(16) NOT NULL,

  FOREIGN KEY(employee_cf) REFERENCES employee(cf)
    ON UPDATE CASCADE
    ON DELETE CASCADE
);

CREATE TABLE message (
  timestamp TIMESTAMP(0) WITHOUT TIME ZONE DEFAULT current_timestamp,
  text TEXT,
  resource_id INTEGER NOT NULL,
  employee_cf CHAR(16) NOT NULL,

  FOREIGN KEY (resource_id) REFERENCES resource(id)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
  FOREIGN KEY (employee_cf) REFERENCES employee(cf)
    ON UPDATE CASCADE
    ON DELETE CASCADE,

  PRIMARY KEY(employee_cf, timestamp)
);

CREATE TABLE session (
  id INTEGER PRIMARY KEY
);

CREATE TABLE shift (
  id SERIAL PRIMARY KEY,
  shift_date DATE NOT NULL,
  employee_cf CHAR(16) NOT NULL,
  session_id INTEGER NOT NULL,
```



```
hour_start SMALLINT NOT NULL,  
hour_end SMALLINT NOT NULL,  
  
FOREIGN KEY(employee_cf) REFERENCES employee(cf)  
    ON UPDATE CASCADE  
    ON DELETE CASCADE,  
FOREIGN KEY(session_id) REFERENCES session(id)  
    ON UPDATE CASCADE  
    ON DELETE CASCADE,  
  
UNIQUE(shift_date, employee_cf),  
UNIQUE(shift_date, session_id),  
  
CHECK(hour_start >= 0 AND hour_start < 24),  
CHECK(hour_end >= 0 AND hour_end < 24),  
CHECK(hour_end > hour_start)  
);  
  
CREATE TABLE maintainer (  
    shift_id INTEGER PRIMARY KEY,  
  
    FOREIGN KEY (shift_id) REFERENCES shift(id)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE  
);  
  
CREATE TABLE dispatcher (  
    shift_id INTEGER PRIMARY KEY,  
  
    FOREIGN KEY (shift_id) REFERENCES shift(id)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE  
);  
  
CREATE TABLE condition (  
    name VARCHAR(50) PRIMARY KEY  
);  
  
CREATE TABLE device (  

```

```
name VARCHAR(50) PRIMARY KEY,
specs TEXT
);

CREATE TABLE task (
  name CHAR(8) PRIMARY KEY,
  description TEXT NOT NULL,
  resource_id INTEGER,
  FOREIGN KEY(resource_id) REFERENCES resource(id)
  ON UPDATE CASCADE
  ON DELETE SET NULL
);

CREATE TABLE groupn (
  title CHAR(4) PRIMARY KEY,
  damage SMALLINT NOT NULL,
  risk SMALLINT NOT NULL,

  CHECK (damage >= 1 AND damage <= 3),
  CHECK (risk >= 1 AND risk <= 3)
);

CREATE TABLE condition_groupn (
  condition_name VARCHAR(50) NOT NULL,
  groupn_title CHAR(4) NOT NULL,
  FOREIGN KEY(condition_name) REFERENCES condition(name)
  ON UPDATE CASCADE
  ON DELETE CASCADE,
  FOREIGN KEY(groupn_title) REFERENCES groupn(title)
  ON UPDATE CASCADE
  ON DELETE CASCADE,

  PRIMARY KEY(condition_name, groupn_title)
);

CREATE TABLE device_groupn (
  device_name VARCHAR(50) NOT NULL,
  groupn_title CHAR(4) NOT NULL,
  FOREIGN KEY(device_name) REFERENCES device(name)
```

```
    ON UPDATE CASCADE
    ON DELETE CASCADE,
FOREIGN KEY(groupn_title) REFERENCES groupn(title)
    ON UPDATE CASCADE
    ON DELETE CASCADE,

PRIMARY KEY(device_name, groupn_title)
);
CREATE TABLE task_groupn (
    task_name CHAR(8) NOT NULL,
    groupn_title CHAR(4) NOT NULL,
    FOREIGN KEY(task_name) REFERENCES task(name)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY(groupn_title) REFERENCES groupn(title)
        ON UPDATE CASCADE
        ON DELETE CASCADE,

    PRIMARY KEY(task_name, groupn_title)
);
CREATE TABLE intervention (
    maintainer_shift_id INTEGER NOT NULL,
    task_name CHAR(8) NOT NULL,
    town_cap CHAR(6) NOT NULL,
    address VARCHAR(40) NOT NULL,
    description TEXT,
    start_at TIMESTAMP(0) WITHOUT TIME ZONE NOT NULL,
    end_at TIMESTAMP(0) WITHOUT TIME ZONE,

    FOREIGN KEY(maintainer_shift_id) REFERENCES maintainer(shift_id)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY(task_name) REFERENCES task(name)
        ON UPDATE CASCADE
        ON DELETE NO ACTION,
```

```
FOREIGN KEY(town_cap) REFERENCES town(cap)
    ON UPDATE CASCADE
    ON DELETE NO ACTION,

PRIMARY KEY(maintainer_shift_id, start_at),

CHECK(end_at > start_at)
);
CREATE TABLE inventory (
    nr INTEGER NOT NULL,
    device_name VARCHAR(50) NOT NULL,
    description TEXT,

    FOREIGN KEY(device_name) REFERENCES device(name)
        ON UPDATE CASCADE
        ON DELETE CASCADE,

    PRIMARY KEY(nr, device_name)
);
CREATE TABLE borrow (
    maintainer_shift_id INTEGER,
    inventory_nr INTEGER NOT NULL,
    inventory_device_name VARCHAR(50) NOT NULL,
    motivation TEXT,
    start_at TIMESTAMP(0) WITHOUT TIME ZONE NOT NULL,
    end_at TIMESTAMP(0) WITHOUT TIME ZONE,

    FOREIGN KEY(inventory_nr, inventory_device_name) REFERENCES
inventory(nr, device_name)
        ON UPDATE CASCADE
        ON DELETE NO ACTION,
    FOREIGN KEY(maintainer_shift_id) REFERENCES maintainer(shift_id)
        ON UPDATE CASCADE
        ON DELETE SET NULL,
```

```
PRIMARY KEY(inventory_nr, inventory_device_name, start_at),

CHECK(end_at > start_at)
);
CREATE TABLE assistance (
  maintainer_shift_id INTEGER,
  ticket SERIAL PRIMARY KEY,
  start_at TIMESTAMP(0) WITHOUT TIME ZONE NOT NULL,
  end_at TIMESTAMP(0) WITHOUT TIME ZONE,

  FOREIGN KEY(maintainer_shift_id) REFERENCES maintainer(shift_id)
    ON UPDATE CASCADE
    ON DELETE SET NULL,

  CHECK(end_at > start_at)
);
CREATE TABLE attendee (
  assistance_ticket INTEGER NOT NULL,
  dispatcher_shift_id INTEGER NOT NULL,

  FOREIGN KEY(assistance_ticket) REFERENCES assistance(ticket)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
  FOREIGN KEY(dispatcher_shift_id) REFERENCES dispatcher(shift_id)
    ON UPDATE CASCADE
    ON DELETE CASCADE,

  PRIMARY KEY(assistance_ticket, dispatcher_shift_id)
);
```

Triggers e Stored Procedure

```
/*
* Controllo se un dipendente inserito come manutentore con un particolare
  shift_id non si ripeta nella tabella other_table
* other_table è un parametro, in modo da generalizzare la function a
  qualsiasi tabella passata in parametro.
```

* Per ora le tabelle di interesse sono: 'dispatcher', 'maintainer'

* Vedere sezione Vincoli Aggiuntivi

*/

```
CREATE OR REPLACE FUNCTION shift_overlap_employee_type ()
  RETURNS trigger AS $$
  DECLARE
    shift_id INTEGER;
    other_table text;
  BEGIN
    other_table = TG_ARGV[0];
    EXECUTE format('SELECT shift_id FROM %I WHERE shift_id = $1',
other_table) INTO shift_id USING NEW.shift_id;
    IF shift_id IS NOT NULL THEN
      RAISE EXCEPTION 'shift_overlap_employee_type(%). id Already
exists', other_table
        USING HINT = 'Please check your shift id on tables';
    END IF;
    RETURN NEW;
  END
$$ LANGUAGE 'plpgsql';
```

/*

* Controllo se la risorsa in questione fa riferimento a un nodo
dell'albero di tipo foglia

*/

```
CREATE OR REPLACE FUNCTION is_resource_a_leaf ()
  RETURNS trigger AS $$
  DECLARE
    resource_tb record;
  BEGIN
    IF NEW.resource_id IS NOT NULL THEN
      SELECT resource.id INTO resource_tb FROM resource WHERE
resource.parent = NEW.resource_id;
      IF resource_tb IS NOT NULL THEN
        RAISE EXCEPTION 'is_resource_a_leaf (%). resource_id is not a
leaf', NEW.resource_id
          USING HINT = 'Please check your resource table';
      END IF;
    END IF;
    RETURN NEW;
  END
```

```
$$ LANGUAGE 'plpgsql';/*
* Controllo se le date rientrano nei turni di un specifico dipendente che
* fa un operazione
* Vedere sezione 'Vincoli aggiuntivi'
* Funzione generica, usata dai triggers per ogni tabella di interesse (es
* intervention)
*/
CREATE OR REPLACE FUNCTION shift_dates_equalities_function()
  RETURNS trigger AS $$
  DECLARE
    _shift record;
  BEGIN
    IF NEW.start_at IS NOT NULL THEN
      SELECT * INTO _shift FROM shift WHERE id =
NEW.maintainer_shift_id
      AND NEW.start_at BETWEEN shift_date + interval '1 hour' *
hour_start AND shift_date + interval '1 hour' * hour_end;
      END IF;

      IF _shift IS NULL THEN
        RAISE EXCEPTION 'shift_dates_equalities_function exception.
Timestamp not valid for the shift'
        USING HINT = 'Please check your shift dates and the table
start_at or end_at timestamps';
      END IF;

      IF NEW.end_at IS NOT NULL THEN
        SELECT * INTO _shift FROM shift WHERE id =
NEW.maintainer_shift_id
        AND NEW.end_at BETWEEN shift_date + interval '1 hour' *
hour_start AND shift_date + interval '1 hour' * hour_end;
        END IF;

        IF _shift IS NULL THEN
          RAISE EXCEPTION 'shift_dates_equalities_function exception.
Timestamp not valid for the shift'
          USING HINT = 'Please check your shift dates and the table
start_at or end_at timestamps';
        END IF;

        RETURN NEW;
      END
    $$ LANGUAGE 'plpgsql';
```

```
CREATE TRIGGER media_is_resource_a_leaf_trigger
  BEFORE INSERT OR UPDATE
  ON attachment
  FOR EACH ROW EXECUTE PROCEDURE is_resource_a_leaf();
CREATE TRIGGER task_is_resource_a_leaf_trigger
  BEFORE INSERT OR UPDATE
  ON task
  FOR EACH ROW EXECUTE PROCEDURE is_resource_a_leaf();
CREATE TRIGGER shift_overlap_employee_maintainer_trigger
  BEFORE INSERT OR UPDATE
  ON maintainer
  FOR EACH ROW EXECUTE PROCEDURE
shift_overlap_employee_type('maintainer');

CREATE TRIGGER shift_overlap_employee_dispatcher_trigger
  BEFORE INSERT OR UPDATE
  ON dispatcher
  FOR EACH ROW EXECUTE PROCEDURE
shift_overlap_employee_type('dispatcher');
CREATE TRIGGER shift_dates_equalities_assistance_trigger
  BEFORE INSERT OR UPDATE
  ON assistance
  FOR EACH ROW EXECUTE PROCEDURE shift_dates_equalities_function();

CREATE OR REPLACE FUNCTION dispatcher_shift_equalities_function()
  RETURNS trigger AS $$
  DECLARE
    _shift record;
  BEGIN

    SELECT shift.* INTO _shift FROM shift, assistance WHERE
NEW.assistance_ticket = assistance.ticket
    AND shift.id = NEW.dispatcher_shift_id
    AND assistance.start_at BETWEEN shift_date + interval '1 hour' *
hour_start AND shift_date + interval '1 hour' * hour_end;

    IF _shift IS NULL THEN
      RAISE EXCEPTION 'dispatcher_shift_equalities_function exception.
Shift or Timestamp not valid'
      USING HINT = 'Please check your shift dates and the table
```



```
start_at or end_at timestamps';
    END IF;

    SELECT shift.* INTO _shift FROM shift, assistance WHERE
NEW.assistance_ticket = assistance.ticket
    AND shift.id = NEW.dispatcher_shift_id
    AND (assistance.end_at IS NULL OR assistance.end_at BETWEEN
shift_date + interval '1 hour' * hour_start
    AND shift_date + interval '1 hour' * hour_end);

    IF _shift IS NULL THEN
        RAISE EXCEPTION 'dispatcher_shift_equalities_function exception.
Shift or Timestamp not valid'
        USING HINT = 'Please check your shift dates and the table
start_at or end_at timestamps';
    END IF;
    RETURN NEW;
END
$$ LANGUAGE 'plpgsql';

CREATE TRIGGER dispatcher_shift_equalities_assistance_trigger
    BEFORE INSERT OR UPDATE
    ON attendee
    FOR EACH ROW EXECUTE PROCEDURE
dispatcher_shift_equalities_function();-- Un manutentore non può fare
altre assistenze fino a quando non ha terminato quella presente
CREATE OR REPLACE FUNCTION maintainer_is_already_occupied_assistance()
    RETURNS trigger AS $$
    DECLARE
        _assurances record;
    BEGIN
        SELECT maintainer_shift_id INTO _assurances FROM assistance WHERE
maintainer_shift_id = NEW.maintainer_shift_id AND end_at IS NULL;
        IF _assurances IS NOT NULL THEN
            RAISE EXCEPTION 'maintainer_is_already_occupied_assistance
exception. The maintaner had to finish others assistance first!'
            USING HINT = 'Please check assistance';
        END IF;
        RETURN NEW;
    END
$$ LANGUAGE 'plpgsql';
```

```
CREATE TRIGGER maintainer_is_already_occupied_assistance_trigger
  BEFORE INSERT
  ON assistance
  FOR EACH ROW EXECUTE PROCEDURE
maintainer_is_already_occupied_assistance();
CREATE TRIGGER
shift_dates_equalities_intervention_trigger
  BEFORE INSERT OR UPDATE
  ON intervention
  FOR EACH ROW EXECUTE PROCEDURE shift_dates_equalities_function();
```

-- Un manutentore non può fare altri interventi (indipendentemente dalla mansione) una volta che ha iniziato un determinato intervento

```
CREATE OR REPLACE FUNCTION maintainer_is_already_occupied_intervention()
  RETURNS trigger AS $$
  DECLARE
    _intervents record;
  BEGIN
    SELECT maintainer_shift_id INTO _intervents FROM intervention WHERE
maintainer_shift_id = NEW.maintainer_shift_id AND end_at IS NULL;
    IF _intervents IS NOT NULL THEN
      RAISE EXCEPTION 'maintainer_is_already_occupied_intervention
exception. The maintaner had to finish others intervention first!'
      USING HINT = 'Please check intervations';
    END IF;
    RETURN NEW;
  END
$$ LANGUAGE 'plpgsql';
```

```
CREATE TRIGGER maintainer_is_already_occupied_intervention_trigger
  BEFORE INSERT
  ON intervention
  FOR EACH ROW EXECUTE PROCEDURE
maintainer_is_already_occupied_intervention();
```

```
CREATE TRIGGER shift_dates_equalities_borrow_trigger
  BEFORE INSERT OR UPDATE
  ON borrow
  FOR EACH ROW EXECUTE PROCEDURE shift_dates_equalities_function();
```

-- Un manutentore non può prendere in prestito un oggetto non ancora restituito.

```
CREATE OR REPLACE FUNCTION maintainer_borrow_not_available_object()
  RETURNS trigger AS $$
  DECLARE
    _borrow record;
  BEGIN
    SELECT maintainer_shift_id INTO _borrow FROM borrow WHERE
maintainer_shift_id = NEW.maintainer_shift_id
      AND end_at IS NULL AND inventory_nr = NEW.inventory_nr AND
inventory_device_name = NEW.inventory_device_name;
    IF _borrow IS NOT NULL THEN
      RAISE EXCEPTION 'maintainer_borrow_not_available_object
exception. The Object is not available!'
      USING HINT = 'Please check objects restitution date';
    END IF;
    RETURN NEW;
  END
$$ LANGUAGE 'plpgsql';
```

```
CREATE TRIGGER maintainer_borrow_not_available_object_trigger
  BEFORE INSERT
  ON borrow
  FOR EACH ROW EXECUTE PROCEDURE
maintainer_borrow_not_available_object();
```

Query di inserimento

```
INSERT INTO typology (type) VALUES ('pdf');
INSERT INTO typology (type) VALUES ('video');
INSERT INTO typology (type) VALUES ('image');
INSERT INTO media (id, source, typology_type) VALUES (1,
'/assets/elettrogeno/gruppo_continuita', 'image');
INSERT INTO media (id, source, typology_type) VALUES (2,
'/assets/manuale', 'pdf');
INSERT INTO resource (id, title, parent) VALUES (1, 'ciclo manutenzione
elettrogeno', NULL);
INSERT INTO resource (id, title, parent) VALUES (2, 'informazioni
osdcontroller', NULL);
```

```
INSERT INTO resource (id, title, parent) VALUES (3, 'informazioni
barriera b680h', NULL);
INSERT INTO resource (id, title, parent) VALUES (4, 'verifiche vista
controllo integrità prova carico', 1);
INSERT INTO resource (id, title, parent) VALUES (5, 'funzionamento
apparecchiature strumentali', 1);
INSERT INTO resource (id, title, parent) VALUES (6, 'integrità chiusure
segregazioni', 4);
INSERT INTO resource (id, title, parent) VALUES (7, 'controllo
danneggiamenti', 4);
INSERT INTO resource (id, title, parent) VALUES (8, 'integrità targhette
segnaletica sicurezza', 4);
INSERT INTO resource (id, title, parent) VALUES (9, 'esame interno
scariche elettriche', 4);
INSERT INTO resource (id, title, parent) VALUES (10, 'infiltrazioni acqua
condensa quadro', 4);
INSERT INTO resource (id, title, parent) VALUES (11, 'batterie avviamento
sistema ricarica', 5);
INSERT INTO resource (id, title, parent) VALUES (12, 'equilibrio fasi
uscita carico', 5);
INSERT INTO attachment (resource_id, media_id) VALUES (2, 2);
INSERT INTO attachment (resource_id, media_id) VALUES (3, 2);
INSERT INTO attachment (resource_id, media_id) VALUES (7, 1);
INSERT INTO attachment (resource_id, media_id) VALUES (9, 1);
INSERT INTO attachment (resource_id, media_id) VALUES (10, 1);
INSERT INTO town (cap, name) VALUES ('41011', 'Campogalliano');
INSERT INTO town (cap, name) VALUES ('41033', 'Concordia sul Secchia');
INSERT INTO town (cap, name) VALUES ('41123', 'Modena');
INSERT INTO town (cap, name) VALUES ('41012', 'Carpi');
INSERT INTO town (cap, name) VALUES ('42124', 'Reggio Emilia');
INSERT INTO town (cap, name) VALUES ('46100', 'Mantova');
INSERT INTO office (id, address, mail, phone, town_cap) VALUES (1, 'A1
AdS Secchia Ovest, 506', 'asmodena@autostrade.it', '800 108 108',
'41123');
```

```
INSERT INTO office (id, address, mail, phone, town_cap) VALUES (2, 'Via
Fernando Fornaciari, 1', 'ascarpi@autostrade.it', '059 668253', '41012');
INSERT INTO office (id, address, mail, phone, town_cap) VALUES (3, 'A1
casello di Reggio Emilia', 'asreggio@autostrade.it', '800 269 269',
'42124');
INSERT INTO employee (cf, name, surname, address, town_cap, office_id)
VALUES ('GRZMTT83G40F230E', 'Matteo', 'Guerzoni', 'Via Donizzetti, 5',
'41012', 2);
INSERT INTO employee (cf, name, surname, address, town_cap, office_id)
VALUES ('SNGMJT80L08F257P', 'Amarjot', 'Singh', 'Via della pace, 18',
'41012', 2);
INSERT INTO employee (cf, name, surname, address, town_cap, office_id)
VALUES ('RSSMRA77P03D150X', 'Mario', 'Rossi', 'Via Mazzini, 4', '41011',
1);
INSERT INTO employee (cf, name, surname, address, town_cap, office_id)
VALUES ('BNCLGU91C23G224S', 'Luigi', 'Bianchi', 'Via Diazzi, 3', '41123',
1);
INSERT INTO employee (cf, name, surname, address, town_cap, office_id)
VALUES ('SRGFRT89C10H223C', 'Sergio', 'Ferretti', 'Via della libertà, 2',
'46100', 3);
INSERT INTO employee (cf, name, surname, address, town_cap, office_id)
VALUES ('PNFFNC84D07I462B', 'Francesco', 'Panofino', 'Strada Martini, 9',
'42124', 1);
INSERT INTO employee (cf, name, surname, address, town_cap, office_id)
VALUES ('LTRCRL94P15B819G', 'Carlo', 'Altero', 'Via martiri della
libertà, 5', '42124', 3);
INSERT INTO contact (number, employee_cf) VALUES ('+39 3425786932',
'GRZMTT83G40F230E');
INSERT INTO contact (number, employee_cf) VALUES ('+39 059 415622',
'GRZMTT83G40F230E');
INSERT INTO contact (number, employee_cf) VALUES ('+39 3319692342',
'SNGMJT80L08F257P');
INSERT INTO contact (number, employee_cf) VALUES ('+39 0642 25428',
'RSSMRA77P03D150X');
```

```
INSERT INTO contact (number, employee_cf) VALUES ('+39 3459611329',
'BNCLGU91C23G224S');
INSERT INTO message ("timestamp", text, resource_id, employee_cf) VALUES
('2018-08-22 17:40:40', 'batterie avviamento sistema ricarica', 11,
'GRZMTT83G40F230E');
INSERT INTO message ("timestamp", text, resource_id, employee_cf) VALUES
('2018-08-22 17:41:11', 'ciclo manutenzione elettrogeno', 1,
'GRZMTT83G40F230E');
INSERT INTO message ("timestamp", text, resource_id, employee_cf) VALUES
('2018-08-22 17:41:18', 'verifiche vista controllo integrità prova
carico', 4, 'GRZMTT83G40F230E');
INSERT INTO message ("timestamp", text, resource_id, employee_cf) VALUES
('2018-08-22 17:41:27', 'integrità chiusure segregazioni', 6,
'GRZMTT83G40F230E');
INSERT INTO message ("timestamp", text, resource_id, employee_cf) VALUES
('2018-08-22 17:41:36', 'equilibrio fasi uscita carico', 12,
'GRZMTT83G40F230E');
INSERT INTO message ("timestamp", text, resource_id, employee_cf) VALUES
('2018-08-22 17:41:42', 'ciclo manutenzione elettrogeno', 1,
'SNGMJT80L08F257P');
INSERT INTO message ("timestamp", text, resource_id, employee_cf) VALUES
('2018-08-22 17:41:45', 'funzionamento apparecchiature strumentali', 5,
'SNGMJT80L08F257P');
INSERT INTO message ("timestamp", text, resource_id, employee_cf) VALUES
('2018-08-22 17:41:48', 'batterie avviamento sistema ricarica', 11,
'SNGMJT80L08F257P');
INSERT INTO session (id) VALUES (1000);
INSERT INTO session (id) VALUES (1001);
INSERT INTO session (id) VALUES (1002);
INSERT INTO session (id) VALUES (1003);
INSERT INTO session (id) VALUES (1004);
INSERT INTO shift (id, shift_date, employee_cf, session_id, hour_start,
hour_end) VALUES (1, '2018-08-19', 'GRZMTT83G40F230E', 1002, 11, 17);
INSERT INTO shift (id, shift_date, employee_cf, session_id, hour_start,
hour_end) VALUES (2, '2018-08-19', 'RSSMRA77P03D150X', 1003, 15, 20);
```

```
INSERT INTO shift (id, shift_date, employee_cf, session_id, hour_start,
hour_end) VALUES (3, '2018-08-19', 'BNCLGU91C23G224S', 1004, 10, 20);
INSERT INTO shift (id, shift_date, employee_cf, session_id, hour_start,
hour_end) VALUES (4, '2018-08-21', 'SNGMJT80L08F257P', 1000, 10, 18);
INSERT INTO shift (id, shift_date, employee_cf, session_id, hour_start,
hour_end) VALUES (5, '2018-08-21', 'RSSMRA77P03D150X', 1001, 5, 13);
INSERT INTO shift (id, shift_date, employee_cf, session_id, hour_start,
hour_end) VALUES (6, '2018-08-21', 'BNCLGU91C23G224S', 1002, 6, 14);
INSERT INTO shift (id, shift_date, employee_cf, session_id, hour_start,
hour_end) VALUES (7, '2018-08-21', 'SRGFRT89C10H223C', 1003, 10, 19);
INSERT INTO shift (id, shift_date, employee_cf, session_id, hour_start,
hour_end) VALUES (8, '2018-08-22', 'GRZMTT83G40F230E', 1000, 5, 13);
INSERT INTO shift (id, shift_date, employee_cf, session_id, hour_start,
hour_end) VALUES (9, '2018-08-22', 'SRGFRT89C10H223C', 1001, 4, 12);
INSERT INTO shift (id, shift_date, employee_cf, session_id, hour_start,
hour_end) VALUES (10, '2018-08-22', 'PNFFNC84D07I462B', 1002, 3, 11);
INSERT INTO maintainer (shift_id) VALUES (1);
INSERT INTO maintainer (shift_id) VALUES (2);
INSERT INTO maintainer (shift_id) VALUES (4);
INSERT INTO maintainer (shift_id) VALUES (6);
INSERT INTO maintainer (shift_id) VALUES (9);
INSERT INTO dispatcher (shift_id) VALUES (3);
INSERT INTO dispatcher (shift_id) VALUES (5);
INSERT INTO dispatcher (shift_id) VALUES (7);
INSERT INTO dispatcher (shift_id) VALUES (8);
INSERT INTO dispatcher (shift_id) VALUES (10);
INSERT INTO condition (name) VALUES ('puntali protetti');
INSERT INTO condition (name) VALUES ('puntali non protetti');
INSERT INTO condition (name) VALUES ('parti scoperte in tensione entro 30
cm');
INSERT INTO condition (name) VALUES ('parti scoperte in tensione oltre 30
cm');
INSERT INTO condition (name) VALUES ('parti scoperte qualsiasi
condizione');
```

```
INSERT INTO device (name, specs) VALUES ('guanti', 'gialli - ad uso
lavorativo');
INSERT INTO device (name, specs) VALUES ('puntali', '0.9m');
INSERT INTO device (name, specs) VALUES ('maschera', NULL);
INSERT INTO device (name, specs) VALUES ('indumenti a norma', NULL);
INSERT INTO task (name, description, resource_id) VALUES ('ELETT_00',
'Consente di dimostrare il corretto funzionamento del gruppo
elettrogeno', 6);
INSERT INTO task (name, description, resource_id) VALUES ('ELETT_01',
'Analisi elettrogeno per identificare eventuali danni', 7);
INSERT INTO task (name, description, resource_id) VALUES ('ELETT_02',
'Controllo del quadro per la verifica dei corretti messaggi di
segnalazione', 8);
INSERT INTO task (name, description, resource_id) VALUES ('ELETT_03',
'Prova interna sul flusso delle scariche elettriche di tensione', 9);
INSERT INTO task (name, description, resource_id) VALUES ('ELETT_04',
'Messa in sicurezza quadro a seguito di infiltrazione acqua condensata',
10);
INSERT INTO task (name, description, resource_id) VALUES ('ELETT_05',
'Riparazione delle batterie per il funzionamento del gruppo', 11);
INSERT INTO task (name, description, resource_id) VALUES ('ELETT_06',
'Bilanciamento delle fasi per il controllo del carico', 12);
INSERT INTO groupn (title, damage, risk) VALUES ('PE00', 3, 2);
INSERT INTO groupn (title, damage, risk) VALUES ('PE01', 3, 1);
INSERT INTO groupn (title, damage, risk) VALUES ('PE02', 3, 3);
INSERT INTO condition_groupn (condition_name, groupn_title) VALUES
('puntali protetti', 'PE00');
INSERT INTO condition_groupn (condition_name, groupn_title) VALUES
('parti scoperte in tensione entro 30 cm', 'PE00');
INSERT INTO condition_groupn (condition_name, groupn_title) VALUES
('puntali protetti', 'PE01');
INSERT INTO condition_groupn (condition_name, groupn_title) VALUES
('parti scoperte in tensione oltre 30 cm', 'PE01');
INSERT INTO condition_groupn (condition_name, groupn_title) VALUES
('puntali non protetti', 'PE02');
```



```
INSERT INTO condition_groupn (condition_name, groupn_title) VALUES
('parti scoperte qualsiasi condizione', 'PE02');
INSERT INTO device_groupn (device_name, groupn_title) VALUES ('guanti',
'PE00');
INSERT INTO device_groupn (device_name, groupn_title) VALUES ('indumenti
a norma', 'PE00');
INSERT INTO device_groupn (device_name, groupn_title) VALUES ('guanti',
'PE01');
INSERT INTO device_groupn (device_name, groupn_title) VALUES ('indumenti
a norma', 'PE01');
INSERT INTO device_groupn (device_name, groupn_title) VALUES ('guanti',
'PE02');
INSERT INTO device_groupn (device_name, groupn_title) VALUES ('puntali',
'PE02');
INSERT INTO device_groupn (device_name, groupn_title) VALUES ('indumenti
a norma', 'PE02');
INSERT INTO task_groupn (task_name, groupn_title) VALUES ('ELETT_05',
'PE00');
INSERT INTO task_groupn (task_name, groupn_title) VALUES ('ELETT_05',
'PE01');
INSERT INTO task_groupn (task_name, groupn_title) VALUES ('ELETT_06',
'PE00');
INSERT INTO task_groupn (task_name, groupn_title) VALUES ('ELETT_06',
'PE01');
INSERT INTO task_groupn (task_name, groupn_title) VALUES ('ELETT_06',
'PE02');
INSERT INTO intervention (maintainer_shift_id, task_name, town_cap,
address, description, start_at, end_at) VALUES (1, 'ELETT_01', '41012 ',
'Autostrada A22, km 7', 'Esecuzione intervento di riparazione e controllo
causa maltempo bollino rosso', '2018-08-19 11:23:00', '2018-08-19
13:18:00');
INSERT INTO intervention (maintainer_shift_id, task_name, town_cap,
address, description, start_at, end_at) VALUES (1, 'ELETT_01', '41012 ',
'Autostrada A22, km 7', 'Esecuzione intervento di riparazione e controllo
```

```
causa maltempo bollino rosso', '2018-08-19 12:23:00', '2018-08-19
15:36:00');
INSERT INTO intervention (maintainer_shift_id, task_name, town_cap,
address, description, start_at, end_at) VALUES (2, 'ELETT_05', '41123 ',
'Autostrada A1, km 4', 'Esecuzione intervento di riparazione e controllo
causa maltempo bollino rosso', '2018-08-19 16:37:00', '2018-08-19
18:24:00');
INSERT INTO intervention (maintainer_shift_id, task_name, town_cap,
address, description, start_at, end_at) VALUES (4, 'ELETT_00', '41123 ',
'Autostrada A1, km 11', 'Controllo di routine', '2018-08-21 11:18:00',
'2018-08-21 11:43:00');
INSERT INTO intervention (maintainer_shift_id, task_name, town_cap,
address, description, start_at, end_at) VALUES (4, 'ELETT_02', '41123 ',
'Autostrada A1, km 11', 'Controllo di routine', '2018-08-21 15:27:00',
'2018-08-21 16:08:00');
INSERT INTO intervention (maintainer_shift_id, task_name, town_cap,
address, description, start_at, end_at) VALUES (6, 'ELETT_03', '41123 ',
'Autostrada A1, km 11', 'Controllo di routine', '2018-08-21 06:33:00',
'2018-08-21 07:24:00');
INSERT INTO intervention (maintainer_shift_id, task_name, town_cap,
address, description, start_at, end_at) VALUES (6, 'ELETT_06', '41123 ',
'Autostrada A1, km 11', 'Controllo di routine', '2018-08-21 12:58:00',
'2018-08-21 13:21:00');
INSERT INTO intervention (maintainer_shift_id, task_name, town_cap,
address, description, start_at, end_at) VALUES (9, 'ELETT_04', '42124 ',
'Autostrada A1, km 53', 'Manutenzione per problemi tecnici rilevati',
'2018-08-22 05:42:00', '2018-08-22 06:51:00');
INSERT INTO intervention (maintainer_shift_id, task_name, town_cap,
address, description, start_at, end_at) VALUES (9, 'ELETT_06', '42124 ',
'Autostrada A1, km 53', 'Regolazione carico del gruppo', '2018-08-22
10:22:00', '2018-08-22 11:13:00');
INSERT INTO inventory (nr, device_name, description) VALUES (1,
'puntali', NULL);
INSERT INTO inventory (nr, device_name, description) VALUES (2,
'puntali', NULL);
```

```
INSERT INTO inventory (nr, device_name, description) VALUES (3,
'puntali', NULL);
INSERT INTO inventory (nr, device_name, description) VALUES (4,
'puntali', NULL);
INSERT INTO inventory (nr, device_name, description) VALUES (1, 'guanti',
NULL);
INSERT INTO inventory (nr, device_name, description) VALUES (2, 'guanti',
NULL);
INSERT INTO inventory (nr, device_name, description) VALUES (3, 'guanti',
NULL);
INSERT INTO inventory (nr, device_name, description) VALUES (4, 'guanti',
NULL);
INSERT INTO inventory (nr, device_name, description) VALUES (1,
'maschera', 'Modello sostituito a causa di danneggiamenti');
INSERT INTO inventory (nr, device_name, description) VALUES (2,
'maschera', NULL);
INSERT INTO inventory (nr, device_name, description) VALUES (1,
'indumenti a norma', NULL);
INSERT INTO inventory (nr, device_name, description) VALUES (2,
'indumenti a norma', NULL);
INSERT INTO borrow (inventory_nr, inventory_device_name,
maintainer_shift_id, motivation, start_at, end_at) VALUES (1, 'guanti',
2, 'DPI minimale per classe di pericolo indicata PE00', '2018-08-19
16:25:00', '2018-08-19 18:38:00');
INSERT INTO borrow (inventory_nr, inventory_device_name,
maintainer_shift_id, motivation, start_at, end_at) VALUES (1, 'indumenti
a norma', 2, 'DPI minimale per classe di pericolo indicata PE00',
'2018-08-19 16:25:00', '2018-08-19 18:38:00');
INSERT INTO borrow (inventory_nr, inventory_device_name,
maintainer_shift_id, motivation, start_at, end_at) VALUES (1, 'guanti',
6, 'DPI minimale per classe di pericolo indicata PE01', '2018-08-21
12:50:00', '2018-08-21 13:30:00');
INSERT INTO borrow (inventory_nr, inventory_device_name,
maintainer_shift_id, motivation, start_at, end_at) VALUES (1, 'indumenti
```

```
a norma', 6, 'DPI minimale per classe di pericolo indicata PE01',
'2018-08-21 12:50:00', '2018-08-21 13:30:00');
INSERT INTO borrow (inventory_nr, inventory_device_name,
maintainer_shift_id, motivation, start_at, end_at) VALUES (1, 'guanti',
9, 'DPI minimale per classe di pericolo indicata PE02', '2018-08-22
10:15:00', '2018-08-22 11:28:00');
INSERT INTO borrow (inventory_nr, inventory_device_name,
maintainer_shift_id, motivation, start_at, end_at) VALUES (1, 'indumenti
a norma', 9, 'DPI minimale per classe di pericolo indicata PE02',
'2018-08-22 10:15:00', '2018-08-22 11:28:00');
INSERT INTO borrow (inventory_nr, inventory_device_name,
maintainer_shift_id, motivation, start_at, end_at) VALUES (1, 'puntali',
9, 'DPI minimale per classe di pericolo indicata PE02', '2018-08-22
10:15:00', '2018-08-22 11:28:00');
INSERT INTO assistance (ticket, maintainer_shift_id, start_at, end_at)
VALUES (1, 1, '2018-08-19 13:04:11', '2018-08-19 13:27:35');
INSERT INTO assistance (ticket, maintainer_shift_id, start_at, end_at)
VALUES (2, 2, '2018-08-19 16:04:11', '2018-08-19 16:27:35');
INSERT INTO assistance (ticket, maintainer_shift_id, start_at, end_at)
VALUES (3, 4, '2018-08-21 10:58:43', '2018-08-21 11:07:12');
INSERT INTO assistance (ticket, maintainer_shift_id, start_at, end_at)
VALUES (4, 4, '2018-08-21 16:18:39', '2018-08-21 16:37:22');
INSERT INTO assistance (ticket, maintainer_shift_id, start_at, end_at)
VALUES (5, 6, '2018-08-21 10:34:25', '2018-08-21 11:13:48');
INSERT INTO assistance (ticket, maintainer_shift_id, start_at, end_at)
VALUES (6, 9, '2018-08-22 06:51:35', '2018-08-22 07:04:06');
INSERT INTO attendee (assistance_ticket, dispatcher_shift_id) VALUES (1,
3);
INSERT INTO attendee (assistance_ticket, dispatcher_shift_id) VALUES (2,
3);
INSERT INTO attendee (assistance_ticket, dispatcher_shift_id) VALUES (3,
5);
INSERT INTO attendee (assistance_ticket, dispatcher_shift_id) VALUES (4,
7);
```

```
INSERT INTO attendee (assistance_ticket, dispatcher_shift_id) VALUES (5, 7);
INSERT INTO attendee (assistance_ticket, dispatcher_shift_id) VALUES (6, 8);
INSERT INTO attendee (assistance_ticket, dispatcher_shift_id) VALUES (6, 10);
```

Query di interrogazione

1. *Dato un id = 4 di una risorsa mostrare i suoi figli, esclusa se stessa*

```
WITH RECURSIVE tree AS (
    SELECT resource.id FROM resource WHERE id = 4
    UNION ALL
    SELECT r.id
    FROM resource r JOIN tree p ON p.id = r.parent
) SELECT * FROM tree WHERE id != 4
```

2. *Selezionare tutte le risorse che hanno almeno 1 allegato*

```
SELECT * FROM attachment JOIN (
    SELECT n.* FROM resource n
    WHERE NOT EXISTS (
        SELECT r.parent FROM resource r WHERE r.parent =
n.id )
) as leaves ON leaves.id = attachment.resource_id
```

3. *Data una mansione trovare tutti i dispositivi di sicurezza ad essa associati*

```
SELECT * FROM task_groupn, groupn, device_groupn
WHERE groupn.title = task_groupn.groupn_title AND
device_groupn.groupn_title = groupn.title AND task_groupn.task_name
= 'ELETT_05'
```

4. *Mostrare tutti i dipendenti che sono stati o solo manutentori o solo centralinisti*

```
(
    SELECT employee_cf FROM shift, dispatcher WHERE shift.id =
dispatcher.shift_id
EXCEPT
    SELECT employee_cf FROM shift, maintainer WHERE
maintainer.shift_id = id
)
UNION
```

```
(
    SELECT employee_cf FROM shift, maintainer WHERE
maintainer.shift_id = id
    EXCEPT
    SELECT employee_cf FROM shift, dispatcher WHERE shift.id =
dispatcher.shift_id
);
```

5. Selezionare l'ufficio col maggior numero di dipendenti

```
SELECT office_id
    FROM employee
    GROUP BY office_id HAVING COUNT(*) >= ALL (SELECT COUNT(*)
FROM employee GROUP BY office_id)
```

6. Data una mansione mostrare tutti i dispositivi di sicurezza

```
SELECT device_name FROM task_groupn as t, groupn as g,
device_groupn as d
    WHERE t.groupn_title = g.title AND d.groupn_title = g.title
AND task_name like 'ELETT_05'
```

7. Tutte le città che non hanno uffici o che non abbiano avuto più di 1 intervento

```
SELECT * FROM town WHERE cap IN (
    SELECT town_cap FROM intervention, town WHERE town.cap =
intervention.town_cap GROUP BY town_cap HAVING COUNT(*) <= 1
) OR cap NOT IN (
    SELECT DISTINCT town_cap FROM office, town WHERE town.cap =
office.town_cap
);
```

8. Elencare la mansione svolta e i nomi dei dipendenti che hanno contribuito a svolgere il maggior numero di interventi nella stessa zona e nella stessa giornata di lavoro

```
SELECT task_name, employee.name, employee.surname FROM (
    SELECT town_cap, shift_date, count(*) as intervention_count
    FROM intervention, shift
    WHERE intervention.maintainer_shift_id = shift.id
    GROUP BY town_cap, shift_date
    ORDER BY intervention_count DESC LIMIT 1
) data, intervention, shift, employee
WHERE intervention.maintainer_shift_id = shift.id
AND shift.employee_cf = employee.cf
AND data.town_cap = intervention.town_cap
```

```
AND data.shift_date = shift.shift_date
```

9. *Per ciascun dipendente riportare il codice fiscale e il numero di minuti dedicato a ciascuna attività tra assistenze (centralinista), assistenze (manutentore) e interventi.*

Si possono notare valori NULL nel caso in cui tale dipendente non abbia svolto mai quella attività.

Per eseguire la seguente query è necessario disporre della seguente funzione

```
CREATE OR REPLACE FUNCTION DATEDIFF (start_t TIMESTAMP, end_t
TIMESTAMP)
RETURNS INT AS $$
  DECLARE
    diff INT = 0;
  BEGIN
    diff = diff + DATE_PART('day', end_t - start_t) * 24;
    diff = diff + DATE_PART('hour', end_t - start_t) * 60;
    diff = diff + DATE_PART('minute', end_t - start_t);
  RETURN diff;
END;
$$ LANGUAGE plpgsql;
```

```
SELECT employee.cf,
  op_intervention.minutes AS time_intervention,
  op_assistance.minutes AS time_assistance,
  op_attendee.minutes AS time_attendee
FROM employee
LEFT JOIN (
  SELECT employee_cf,
    SUM(DATEDIFF(intervention.start_at, intervention.end_at)) AS
minutes
  FROM intervention, shift
  WHERE intervention.maintainer_shift_id = shift.id
  AND intervention.end_at IS NOT NULL
  GROUP BY employee_cf
) AS op_intervention ON op_intervention.employee_cf = employee.cf
LEFT JOIN (
  SELECT employee_cf,
    SUM(DATEDIFF(assistance.start_at, assistance.end_at)) AS
minutes
```

```
FROM assistance, shift
WHERE assistance.maintainer_shift_id = shift.id
AND assistance.end_at IS NOT NULL
GROUP BY employee_cf
) AS op_assistance ON op_assistance.employee_cf = employee.cf
LEFT JOIN (
    SELECT shift.employee_cf,
           SUM(DATEDIFF(assistance.start_at, assistance.end_at)) AS
minutes
    FROM assistance, attendee, shift
    WHERE attendee.assistance_ticket = assistance.ticket
    AND assistance.end_at IS NOT NULL
    AND attendee.dispatcher_shift_id = shift.id
    GROUP BY employee_cf
) AS op_attendee ON op_attendee.employee_cf = employee.cf
```

Query di modifica

Aggiornare un intervento di un particolare manutentore con una data di fine. Questo update darà un errore poichè non combacia il turno del manutentore con l'orario inserito (vedere Vincoli aggiuntivi)

```
UPDATE intervention SET end_at = '2019-01-01 10:00:00' WHERE
maintainer_shift_id = 1
```

Modificare l'indirizzo a 'Via Baldassare Peruzzi' del dipendente con codice fiscale SNGMJT80L08F257P

```
UPDATE employee SET address = 'Via Baldassare Peruzzi' WHERE cf =
'SNGMJT80L08F257P';
```

Aggiornare la descrizione del dispositivo 'guanti' con 'verdi - ad uso lavorativo'

```
UPDATE device SET specs = 'verdi - ad uso lavorativo' WHERE name =
'guanti';
```

Query di eliminazione

Eliminare tutti gli interventi riferiti alla mansione 'ELETT_01'

```
DELETE FROM intervention WHERE task_name = 'ELETT_01';
```

Eliminare tutti gli allegati che si riferiscono alle risorse che hanno nel titolo la parola 'informazioni'


```
DELETE FROM attachment WHERE resource_id  
IN (SELECT resource.id FROM resource WHERE resource.title LIKE  
'%informazioni%');
```

Eliminare tutte le città dove non c'è stato neanche un intervento e che non abbiano sedi di riferimento

```
DELETE FROM town WHERE cap NOT IN (  
    SELECT DISTINCT town_cap FROM intervention, town WHERE town.cap =  
intervention.town_cap  
    UNION  
    SELECT DISTINCT town_cap FROM office, town WHERE town.cap =  
office.town_cap  
    UNION  
    SELECT DISTINCT town_cap FROM employee WHERE town.cap =  
employee.town_cap  
);
```

Bibliografia

<https://softwareengineering.stackexchange.com/questions/158534/pros-and-cons-of-holding-all-the-business-logic-in-stored-procedures-in-web-app>

https://en.wikipedia.org/wiki/Multitier_architecture

<https://www.marklogic.com/blog/relational-databases-scale/>

<https://stackoverflow.com/questions/43922915/business-logic-in-stored-procedures-vs-middle-layer>

<http://www.cs.toronto.edu/~mashiyat/csc309/Lectures/Web%20App%20Architectures.pdf>

<https://www.postgresql.org/docs/9.2/static/plpgsql-trigger.html>

<https://www.postgresql.org/docs/9.1/static/sql-createfunction.html>

<https://www.postgresql.org/docs/9.1/static/xfunc.html>

https://www.gnu.org/software/make/manual/html_node/Quick-Reference.html

<https://stackoverflow.com/questions/9657048/constraints-and-assertions-in-postgresql>

<http://dolly.fim.unimore.it/2017/SezioneBasiDiDati>

<https://www.postgresql.org/docs/9.1/static/plpgsql.html>

<https://www.postgresql.org/docs/current/static/plpgsql-statements.html#PLPGSQL-STATEMENTS-SQL-ONEROW>

<http://www.postgresqltutorial.com/postgresql-recursive-query/>