

# Progettazione della base di dati per le gestione di una casa di riposo

Basi di Dati 2022/2023

Beatrice Musizza

January 23, 2023

# 1 Raccolta e analisi dei requisiti

## 1.1 Studio della richiesta

Si vuole progettare una base di dati per la gestione di una casa di riposo privata, contenente informazioni sugli ospiti e il personale della struttura:

- Per ogni ospite: il codice fiscale, il nome, il cognome, la data di nascita, la città, la provincia di nascita. Va registrata anche la stanza in cui è ricoverato all'interno della struttura e il suo livello di autosufficienza (indicato da una scala su tre valori). Si vuole anche tener traccia del luogo da cui ogni ospite arriva: un'ospite può essere accolto dal suo domicilio privato (in questo caso va registrato l'indirizzo e la città) o da un'altra struttura (di cui va salvato il nome, il codice regionale identificativo, l'indirizzo, la città e la data di apertura).

Per ogni ospite, si tiene traccia anche della data in cui è stato accolto e di tutti i ricoveri effettuati in ospedale durante il periodo passato in struttura. In particolare, ogni qualvolta l'ospite fosse ricoverato al di fuori della struttura, va registrato il nome dell'ospedale e la città in cui si trova, la data di ricovero e l'eventuale data di dimissioni.

Va tenuto traccia anche della dieta seguita dagli ospiti. Si tiene traccia del tipo (regolare, ipoglicidica, iposodica, ipoproteica o vegetariana), della data di inizio della dieta e dell'eventuale data di fine. Va registrato anche se la dieta è da somministrare intera o frullata (in caso di disfagia, difficoltà a deglutire).

- Per ogni parente in contatto con l'ospite: il nome, il cognome, la data di nascita, la città, la provincia di residenza, il grado di parentela, il numero di telefono e tutte le visite effettuate.

Per ogni visita fatta dai parenti, si vuole tener traccia della data, dell'ora di inizio della visita e del portinaio che ha effettuato il controllo Green Pass (necessario per accedere alle strutture mediche residenziali).

Nella struttura lavorano medici, infermieri, Operatori socio-sanitari, fisioterapisti, personale amministrativo e portinai.

- I medici lavoranti nella struttura non sono dipendenti diretti di quest'ultima, si tratta di medici di famiglia dipendenti dell'Azienda Sanitaria (ASUGI). Va tenuto traccia del loro codice fiscale, nome, cognome, data di nascita, la città, la provincia di nascita e del numero di iscrizione all'ordine con relativa data di iscrizione.

Essi hanno in carico uno o più ospiti all'interno della struttura, a cui fanno visita almeno una volta al mese, e comunque su richiesta degli infermieri. Si vuole tener traccia del numero di ospiti in carico per ogni medico e delle date di tutte le visite effettuate. I medici effettuano al massimo una visita al giorno per ospite.

I medici sono, inoltre, gli unici a poter effettuare diagnosi (durante la visita) e prescrivere i farmaci (a seguito della diagnosi).

Per ogni diagnosi, si tiene traccia del medico che la ha effettuata, dell'ospite, della data, della parte del corpo interessata, del numero identificativo di referto associato e degli eventuali farmaci prescritti a seguito della visita.

Dei farmaci prescritti ad ogni ospite, si vuole tener traccia del codice del farmaco, del quantitativo prescritto, della cadenza (ogni quante ore va somministrato il farmaco) e del tipo di prescrizione (fissa o condizionata).

I medici sono incaricati anche alla somministrazione del vaccino COVID. Si vuole tener traccia di tutte le dosi eseguite a tutti gli ospiti e della data.

- Per ogni infermiere che lavora nella struttura: il codice fiscale, il nome, il cognome, la data di nascita, la città, la provincia e di nascita, il tipo di contratto (a tempo determinato o a tempo indeterminato) e il numero di iscrizione all'albo professionale, con relativa data di iscrizione.

Gli infermieri all'interno della struttura sono incaricati di stilare il 'piano operativo fisso assistenziale'. Si tratta di un piano personale e specifico per ogni ospite, composto da diverse mansioni. Di

questi va tenuto traccia del tipo: ‘misurazione parametro vitale’ o ‘medicazione’. Delle misurazioni dei parametri vitali va registrato il parametro (pressione sanguigna, peso, temperatura o frequenza cardiaca), delle medicazioni va indicata la parte del corpo interessata. Le mansioni possono essere a cadenza giornaliera o settimanale. Ogni mansione ha un codice identificativo.

Gli infermieri sono anche addetti alla somministrazione dei farmaci. Di tutti i farmaci che vengono somministrati si vuole tener traccia: dell’infermiere che ha somministrato il farmaco, dell’ospite a cui è stato somministrato, del codice del farmaco, della data e dell’ora della somministrazione.

Gli infermieri effettuano, infine, i tamponi di controllo COVID a tutti i dipendenti della struttura (OSS, infermieri, fisioterapisti e amministrativi) e a tutti gli ospiti.

- Per ogni operatore sociosanitario (OSS) che lavora nella struttura: il codice fiscale, il nome, il cognome, la data di nascita, la città, la provincia di nascita, il tipo di contratto (a tempo determinato con data di scadenza o a tempo indeterminato). Gli OSS sono incaricati di svolgere il ‘piano operativo fisso assistenziale’, esso prevede almeno una mansione. Ogni mansione effettuata è detta prestazione e ne va registrata la data e l’ora.

- Per ogni fisioterapista: il codice fiscale, il nome, il cognome, la data di nascita, la città di nascita, il tipo di contratto (a tempo determinato o a tempo indeterminato) e il numero di iscrizione all’albo professionale, con data di iscrizione.

I fisioterapisti si occupano di assegnare gli ausili agli ospiti. Per ogni ausilio va indicato il tipo (carrozzina, deambulatore, materasso ad aria, stampelle o bastone), la data da cui l’ospite lo ha in uso, il tipo di contratto (se è in comodato d’uso alla struttura, di proprietà della struttura o dell’ospite) e il numero identificativo di inventario.

- Per ogni dipendente amministrativo: codice fiscale, il nome, il cognome, la data di nascita, la città, la provincia di nascita, il tipo di contratto (a tempo determinato o a tempo indeterminato).

Va tenuto traccia di ogni pagamento mensile della retta effettuato dagli ospiti, dell’importo, della data, del numero di fattura e dell’operatore amministrativo che ha emesso la fattura associata.

- Per ogni portinaio: codice fiscale, il nome, il cognome, la data di nascita, la città, la provincia di nascita, il tipo di contratto (a tempo determinato con data di scadenza o a tempo indeterminato). Essi sono addetti al controllo Green Pass dei parenti per le visite.

## 1.2 Ulteriori specifiche

Vengono considerate altre specifiche necessarie alla progettazione

- Si tiene traccia dello storico degli ospiti, cioè anche degli ospiti deceduti. Nel caso di decesso, si registra la data di morte.
- Il portinaio incaricato al controllo Green Pass per le visite parente è uno per turno. Di conseguenza non è possibile iniziare nello stesso giorno, alla stessa ora più visite.
- Un medico può far visita a un'ospite e prescrivergli i farmaci solo se questo gli è in carico.
- È possibile che un ospite non abbia mai avuto bisogno di un farmaco.
- I tamponi di controllo vengono effettuati dagli infermieri con cadenza non fissa: essa viene decisa in base all'andamento dell'epidemia e all'eventuale presenza di casi in struttura. Per questo motivo, va tenuta traccia non solo dell'esito ma anche data in cui è stato effettuato ogni tampone.
- Per quello che riguarda il 'piano operativo fisso assistenziale', esso viene aggiornato mensilmente. È necessario tenerne lo storico per aver traccia delle prestazioni effettuate dagli OSS e della loro efficacia.
- Ogni 'piano operativo fisso assistenziale' viene identificato da un codice.
- La stessa mansione può essere prescritta all'interno di due 'piani operativi fissi assistenziali' con due cadenze diverse a due ospiti diversi. La cadenza della singola mansione può anche cambiare per lo stesso ospite in due piani riferiti a due mesi diversi (ad esempio, potrebbe diventare necessario misurare la frequenza cardiaca ogni giorno in seguito all'insorgere di una patologia).
- È presente il vincolo secondo cui i farmaci somministrati dagli infermieri devono essere prima stati prescritti dai medici.
- Non vi è una dipendenza funzionale tra la data di iscrizione all'ordine (per i medici) o all'albo (per infermieri e fisioterapisti) e il numero di iscrizione all'ordine o all'albo. Può infatti accadere che i numeri vengano cambiati nel corso del tempo per dinamiche organizzative interne agli ordini di riferimento.
- Lo stesso ausilio fisioterapico non può essere usato da due ospiti contemporaneamente. È, inoltre, necessario tener traccia dello storico dell'utilizzo degli ausili.

## 2 Requisiti strutturali

### 2.1 Glossario

Per rendere piu' chiaro il significato dei termini chiave relativi ai dipendenti e le relazioni che i requisiti strutturati definiscono tra essi viene fornito un glossario esplicativo:

Termine	Informazioni	Mansioni	Termini correlati
Ospiti	Dati anagrafici, locazione nella struttura, livello autosufficienza, informazioni accoglienza, ricoveri esterni, visite ricevute, dieta, tamponi COVID ricevuti, vaccini COVID ricevuti		Luogo provenienza, Ricovero, Ospedale, Dieta, Visita, Tampone COVID ricevuto, Vaccino COVID ricevuto
Parenti	Dati anagrafici, parentela, visite effettuate		Ospite, Visita, Portinaio
Medici	Dati anagrafici, dati contratto lavoro, ospiti in carico, tamponi COVID ricevuti	Visite almeno mensili agli ospiti, diagnosi, prescrizione farmaci, somministrazione vaccino COVID	Ospite, Visita Medica, Diagnosi, Prescrizione Farmaco, Vaccino COVID somministrato, Tampone COVID ricevuto
Infermiere	Dati anagrafici, dati contratto lavoro, tamponi COVID ricevuti, vaccini COVID ricevuti	Stilare 'piano operativo fisso assistenziale', somministrazione farmaci, somministrazione tamponi di controllo COVID	Piano assistenza, Mansioni, Somministrazione Farmaco, Vaccino COVID ricevuto, Tampone COVID ricevuto, Tampone COVID somministrato
OSS	Dati anagrafici, dati contratto lavoro, tamponi COVID ricevuti, vaccini COVID ricevuti	Svolgere 'piano operativo fisso assistenziale'	Piano assistenza, Prestazione, Tampone COVID ricevuto, Vaccino COVID ricevuto
Fisioterapista	Dati anagrafici, dati contratto lavoro, tamponi COVID ricevuti, vaccini COVID ricevuti	Gestione ausili	Ausilio, Tampone COVID ricevuto, Vaccino COVID ricevuto
Dipendente amministrativo	Dati anagrafici, dati contratto lavoro, tamponi COVID ricevuti, vaccini COVID ricevuti	Emissione fatture	Pagamento, Tampone COVID ricevuto, Vaccino COVID ricevuto
Portinaio	Dati anagrafici, dati contratto lavoro, tamponi COVID ricevuti, vaccini COVID ricevuti	Controllo Green Pass parenti	Parenti, Tampone COVID ricevuto, Vaccino COVID ricevuto

### 3 Requisiti operazionali

Verranno di seguito descritte alcune operazioni significative. La potenzialità della base di dati che si vuole progettare è la multifunzionalità. Essa, infatti, è pensata per la consultazione e la modifica da parte di diverse figure professionali: ciascuno deve essere in grado di poter aggiungere e/o monitorare le informazioni di suo interesse. Questo perché all'interno della casa di riposo è fondamentale lo scambio di informazioni tra diverse entità. Diverse figure professionali devono avere a disposizione diverse indicazioni fornite da altri soggetti. Ad esempio, l'OSS deve avere accesso ai piani stilati dagli infermieri, mentre gli infermieri devono poter verificare le prescrizioni mediche.

Si vanno quindi ad indicare sia delle operazioni generiche che delle operazioni specifiche per una singola figura professionale.

- OPERAZIONE 1 : Si decide di organizzare una festa di compleanno ogni ultimo giorno del mese, dove vengono festeggiati tutti coloro che hanno compiuto gli anni nel mese precedente. Viene quindi richiesto di restituire la lista dei compleanni di febbraio.

- OPERAZIONE 2 : A fini statistici restituire gli ospiti attualmente presenti in struttura, gli ospiti accolti nell'ultimo anno e gli ospiti deceduti nell'ultimo anno.

- OPERAZIONE 3: Si vuole risalire agli ospiti sanissimi, quelli mai ricoverati in ospedale da quando sono ospiti nella casa di riposo e che attualmente non hanno prescrizioni di farmaci in corso).

- OPERAZIONE 4: Si chiede di restituire gli ospiti che non sono in stanza con un ospite con il loro stesso nome.

- OPERAZIONE 5: Si chiede di restituire gli ospiti attualmente positivi al COVID.

- OPERAZIONE 6: Un ospite è positivo al COVID, per poter avvisare tutti coloro che sono stati in struttura a fare visita a un parente, è richiesto il nome e il numero di telefono dei parenti che sono stati in visita delle ultime due settimane

- OPERAZIONE MEDICO : Controllare di aver effettuato almeno una visita a tutti gli ospiti in carico nell'ultimo mese.

- OPERAZIONE INFERMIERE : Restituire per un dato ospite i farmaci attualmente prescritti, in modo da sapere cosa va somministrato.

- OPERAZIONE INFERMIERE : Restituire ospiti e dipendenti che non hanno effettuato tamponi COVID nell'ultimo mese.

- OPERAZIONE OSS: Per ogni ospite in struttura, restituire le prestazioni da fare: le mansioni giornaliere non ancora svolte nelle ultime 24 ore, le mansioni settimanali non ancora svolte negli ultimi 7 giorni.

- OPERAZIONE FISIOTERAPISTA: Per ogni ospite in struttura ottenere il numero di ausili prescritti attualmente e in passato.

- OPERAZIONE OPERATORE AMMINISTRATIVO: Per ogni ospite in struttura, quanto ha pagato nel corso del suo soggiorno in struttura.

## 4 Progettazione concettuale

### 4.1 Costruzione dello schema Entità-Relazione

Per la progettazione dello schema E-R 'è stata utilizzata la tecnica bottom-up: per prima cosa si isolano le richieste del testo in concetti indipendenti e si costruisce uno schema entità-relazione per ciascuno di essi. I singoli diagrammi sono successivamente integrati in un unico schema che va a soddisfare tutte le specifiche.

#### 4.1.1 Ospiti

Commenti e spiegazioni riferiti allo schema concettuale:

- Poiché è richiesto di mantenere traccia di tutte le informazioni anche degli ospiti deceduti, si sceglie di creare una generica entità 'Ospite' che registri tutte le informazioni e le relazioni di tutti gli ospiti e di introdurre una gerarchia che divide 'Ospite In Vita' da 'Ospite Deceduto', in modo da poter impostare dei vincoli illustrati in seguito.

- Per ciò che riguarda l'entità 'Ospite', si è deciso di considerare come attributi i dati di tipo anagrafico (il codice fiscale, che risulta anche identificatore, il nome, il cognome), la stanza in cui è ricoverato all'interno della struttura e il suo livello di autosufficienza. La nascita viene modellata tramite una relazione e creando un'entità 'Città', che verrà utilizzata anche per la locazione del luogo di provenienza e da altre entità. La relazione nascita tiene perciò conto dell'attributo 'Data di Nascita'.

La gerarchia di 'Ospite' permette di garantire l'inserimento della data di decesso solo per gli ospiti deceduti.

- Per l'accoglienza, essa viene considerata unica e modellata tramite una relazione con l'entità 'Luogo di provenienza'. Questa entità va poi suddivisa tramite una gerarchia in modo da tener traccia dei diversi attributi in base al tipo di luogo.

- Per ciò che riguarda i ricoveri in ospedale, dal momento in cui si vuole tener traccia di tutto il loro storico, non è possibile la formalizzazione tramite una singola relazione con l'entità 'Ospedale'. Si è deciso perciò di reificare e creare l'entità 'Ricovero passato', in relazione con 'Ospedale' e 'Ospite'. In questo modo si va registrare tutto lo storico dei ricoveri. Ogni ospedale è univocamente identificato da nome e città dove si trova. Si assume che l'ospite non possa essere ricoverato nello stesso posto due volte in un giorno.

La gerarchia che prevede 'Ospite in Vita' permette di registrare l'eventualità che l'ospite sia attualmente ricoverato solo per gli ospiti in vita tramite la relazione 'Ricovero Attuale' tra l'entità 'Ospite In Vita' e 'Ospedale'.

- Un discorso analogo vale per la dieta: dal momento in cui si vuole tener traccia dello storico, non è possibile modellarla tramite una singola relazione. Si crea l'entità 'Dieta passata'. Tramite la relazione 'Dieta Attuale' si tiene, inoltre, traccia della dieta in corso, questa relazione è possibile solo per ospiti in vita.

Si è deciso di sfruttare la relazione 'Dieta Attuale' anche per registrare il tipo di somministrazione richiesta (intera o frullata). Poiché la relazione 'Dieta Attuale' è possibile solo per l'entità 'Ospite In Vita', nel momento del decesso sarà necessario eliminare questa relazione. Per questo motivo verranno perse le informazioni a 'Somministrazione' per gli ospiti deceduti, ma ciò non risulta essere un problema per la struttura.

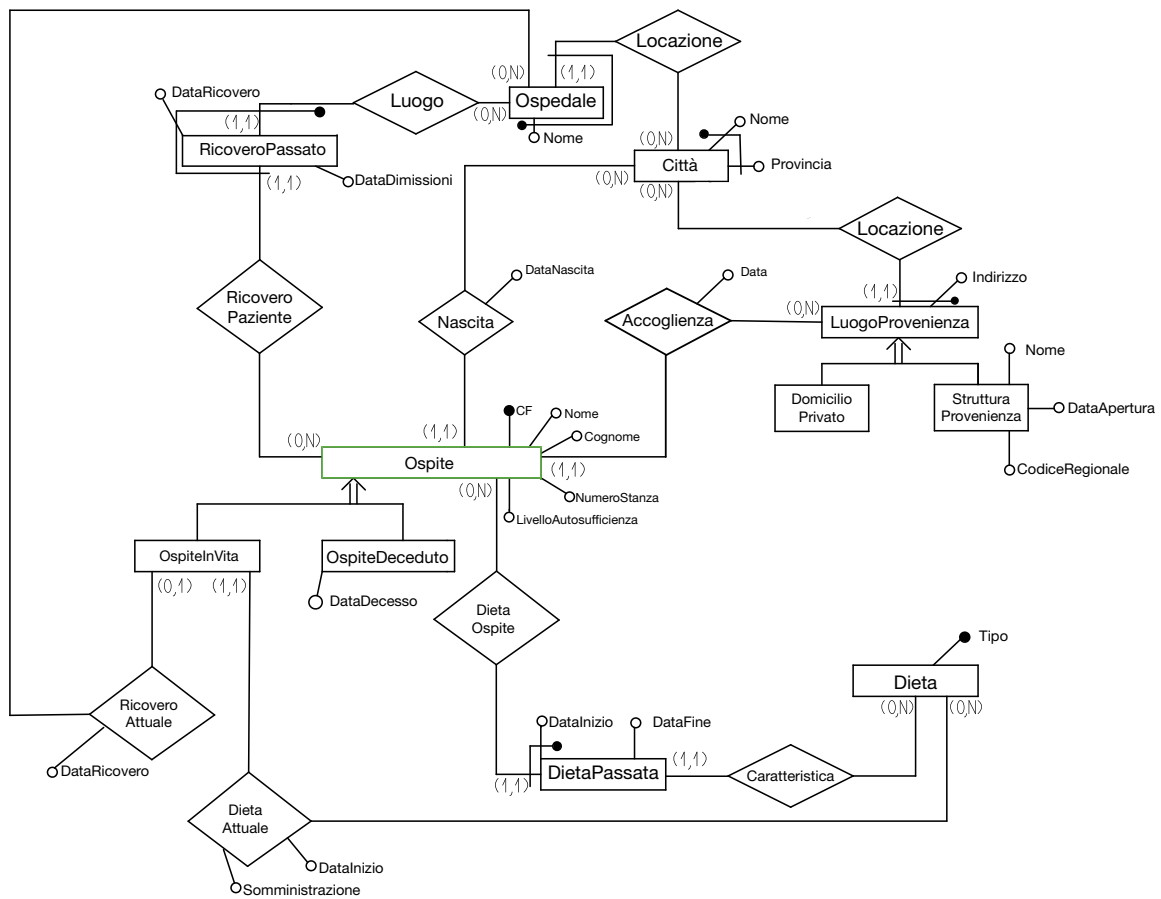


Figure 1: Schema concettuale Ospiti



### 4.1.2 Parenti

Commenti e spiegazioni riferiti allo schema concettuale:

- Per ciò che riguarda l'entità 'Parente', si è impostato come attributi i dati di tipo anagrafico (il codice fiscale, il nome, il cognome e la data di nascita). Per la residenza si utilizza una relazione con l'entità 'Città'.
- Il parente è in relazione con 'Ospite' tramite 'Parentela' che possiede l'attributo 'Grado'.
- Poiché potrebbero esserci più di visite tra lo stesso ospite e parente, si decide di reificare l'entità 'Visita'. La visita è in relazione anche con 'Portinaio' (per il controllo del Green Pass). L'entità 'Portinaio' verrà elaborata in seguito, essendo una sottocategoria dell'entità 'Lavoratori'.

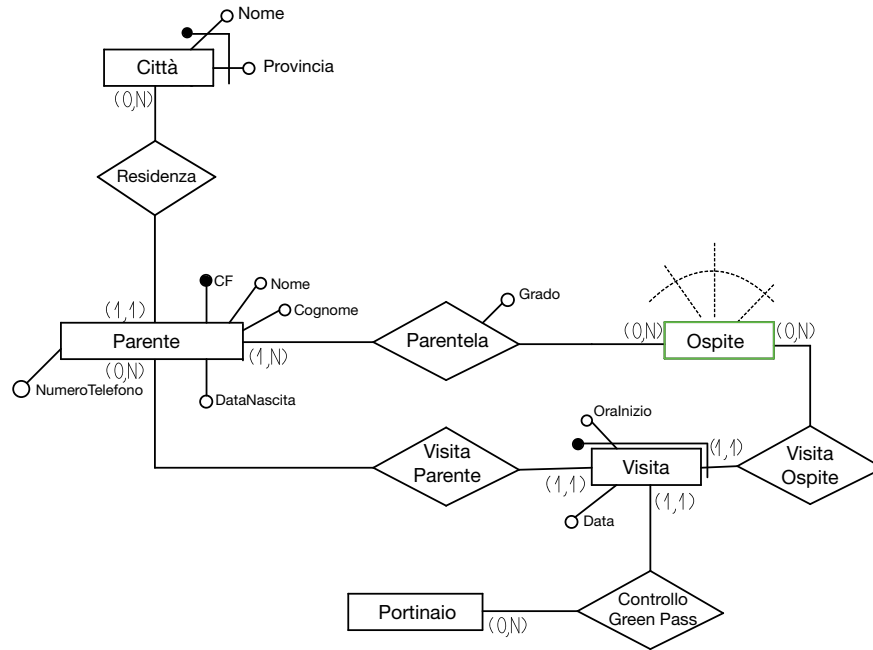


Figure 2: Schema concettuale Parenti

### 4.1.3 Lavoratori

Per tutti i lavoratori vengono salvati gli stessi dati anagrafici e per molti anche gli stessi dati riguardo il contratto. Emerge anche che per tutti i lavoratori sussiste la necessità di registrare i tamponi COVID e i vaccini COVID ricevuti. Si è quindi deciso di creare una doppia gerarchia. Viene creata l'entità 'Lavoratore', con attributi anagrafici e relazioni riferite ai tamponi e ai vaccini e due sottoentità: 'Medici' e 'Dipendenti'. Solo questa seconda entità prevede l'attributo riguardo il tipo di contratto. A sua volta il personale della struttura viene diviso in infermieri, Operatori socio-sanitari, fisioterapisti, personale amministrativo e portinai.

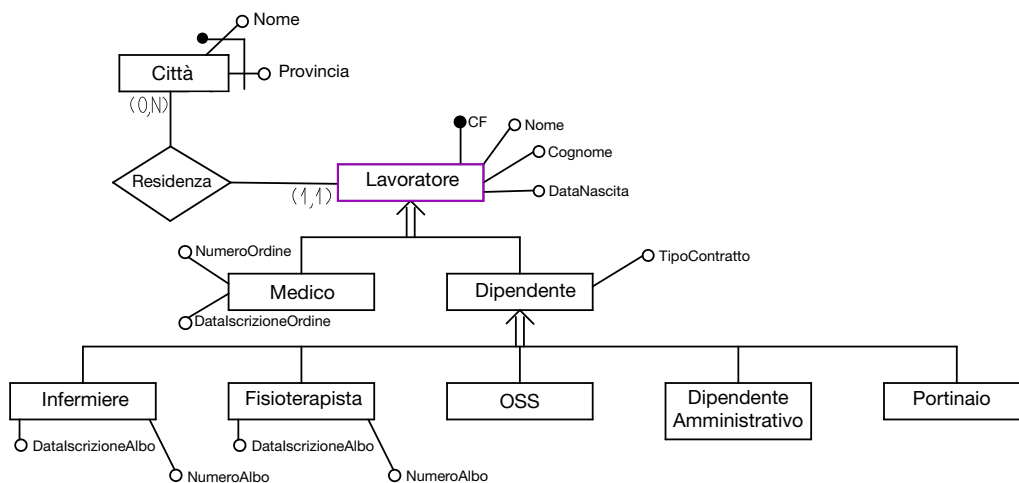


Figure 3: Schema concettuale Lavoratori

#### 4.1.4 Medici

Commenti e spiegazioni riferiti allo schema concettuale:

- Poiché non vi è una dipendenza funzionale tra la data di iscrizione all'ordine e il numero di iscrizione essi vengono entrambi schematizzati come attributi dell'entità 'Medico'.
- Per ciò che riguarda i medici, la principale difficoltà è data dall'assicurare il vincolo che un medico possa far visita, fare diagnosi e prescrivere farmaci solo agli ospiti che gli sono in carico. Creando una semplice relazione 'In Carico' tra medico e ospite si potrebbe andare a modellare le visite tramite un'entità 'Visita' in relazione tra 'Medico' e 'Ospite'. Così facendo non ci si assicura, però, che la visita tra un dato medico e un dato ospite avvenga solo se tra questi due è attiva la relazione 'In Carico'. Per risolvere questo problema è possibile reificare la relazione 'In Carico' rendendola un'entità ('Rapporto Medico'). Così facendo, l'entità 'Visita' viene messa in relazione diretta con l'entità 'Rapporto Medico' permettendo così di creare una reference tra le visite e i rapporti. Tramite opportune relazioni e identificatori esterni, si va poi a creare un riferimento anche tra il 'Rapporto Medico', le diagnosi e le prescrizioni di farmaci. Poiché i medici effettuano al massimo una visita al giorno per ospite, queste vengono identificate dalla data e dal 'Rapporto Medico' a cui sono associate.
- Si noti che non è necessario inserire l'attributo 'Data' per ciò che riguarda le diagnosi, essendo queste univocamente collegate a una visita, di cui viene registrata la data.
- Per quanto riguarda i farmaci si decide di non creare un'entità farmaco con attributo 'Codice Farmaco', poiché non è di interesse tener traccia dei farmaci in sé, quanto delle prescrizioni. Viene perciò creato semplicemente un attributo 'Codice Farmaco' dell'entità 'Prescrizione Farmaco'.
- Per evitare di creare ridondanza e inconsistenza (causata da inserimenti errati o parziali) si decide di non creare un attributo per il numero ospiti in carico a ogni medico. Questo dato è, infatti, facilmente ricavabile dal numero relazioni 'Rapporto Medico' di ogni medico.
- Per ciò che riguarda il vaccino COVID, poiché uno stesso ospite o lavoratore può ricevere più dosi dallo stesso medico, vengono create le entità 'Vaccino Ospite' e 'Vaccino Lavoratore' per evitare la multi-relazione.

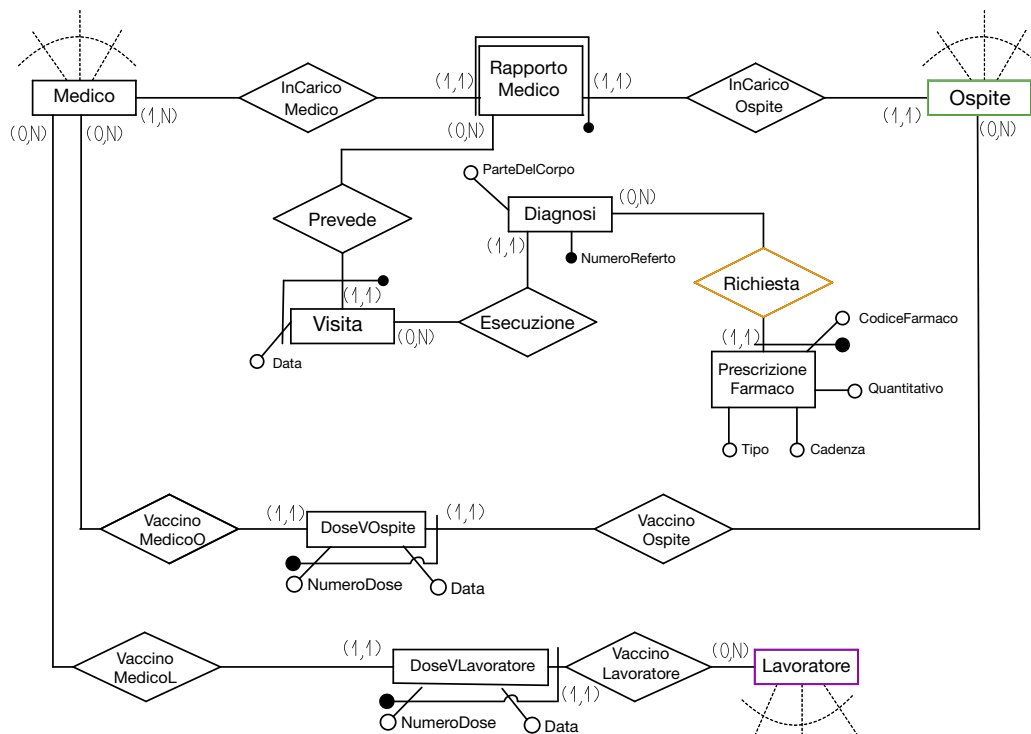


Figure 4: Schema concettuale Medici

#### 4.1.5 Infermieri ed OSS

Commenti e spiegazioni riferiti allo schema concettuale:

- Per ciò che riguarda gli infermieri, si noti che la somministrazione dei farmaci viene modellata tramite una relazione del tipo Kind-Of 'Giustifica', andando a vedere la 'Somministrazione farmaco' come entità concreta a fronte dell'esistenza dell'entità astratta 'Prescrizione farmaco'. In questo modo le somministrazioni avvengono solo vincolate a una prescrizione medica.
- Si assume che in un giorno un lavoratore o un ospite possano effettuare al massimo un tampone con lo stesso infermiere, nel caso in cui sia necessaria una seconda verifica, infatti, se ne occupa un altro infermiere.
- Per le prestazioni (attuazione delle mansioni) degli OSS, si è deciso di creare due entità direttamente collegate ai due diversi tipo di mansioni, in modo da garantire che ogni anche prestazione possa essere alternativamente di tipo Misurazione o Medicazione. Non risulta necessario creare un'entità generica 'Prestazione'.

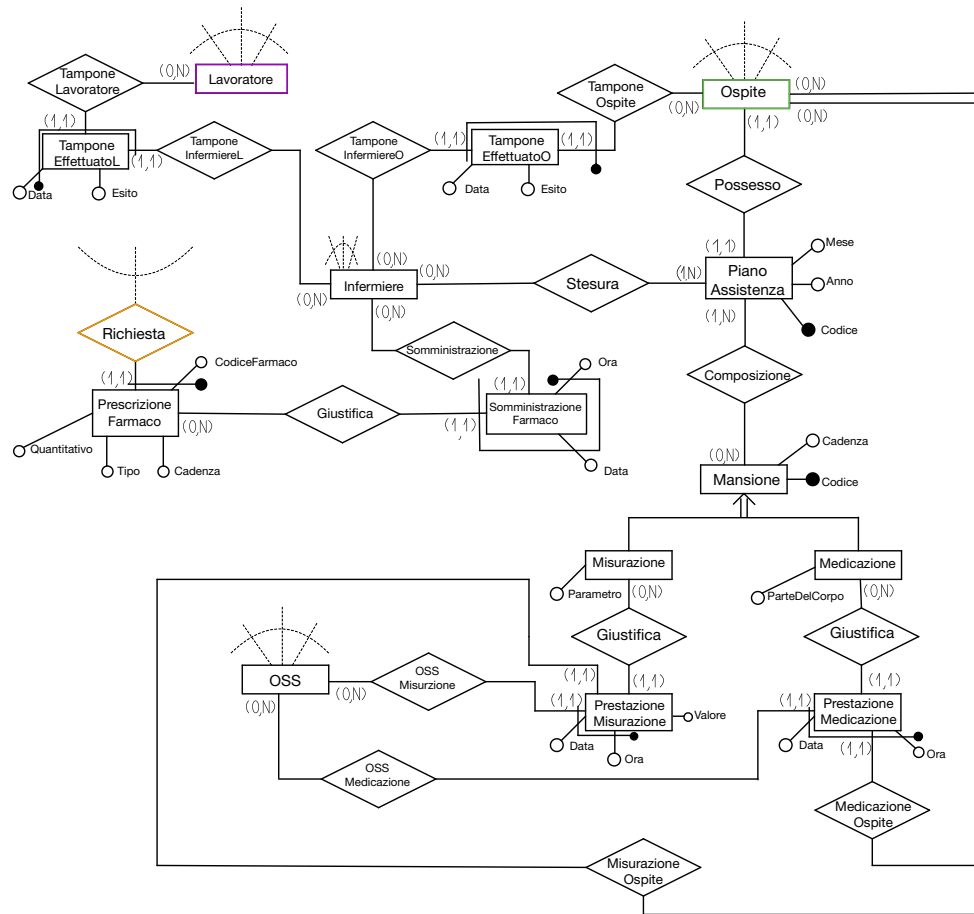


Figure 5: Schema concettuale Infermieri ed OSS

#### 4.1.6 Fisioterapisti e Dipendenti amministrativi

Commenti e spiegazioni riferiti allo schema concettuale:

- Viene reificato 'Assegnazione Ausilio' per poter registrare gli storici. La situazione è simile a quelle delle diete e dei ricoveri, per ciò che riguarda le assegnazioni attuali si crea una relazione solo con l'entità 'OspiteInVita'.
- Per evitare eccessiva complessità, si omettono le entità 'Ausili' e 'Fatture', dal momento che non è di interesse avere tabelle contenenti informazioni solo su questi, vengono posti come attributi.
- Poiché i pagamenti sono mensili, si assume che per ogni mese ad ogni ospite venga emessa una sola fattura.

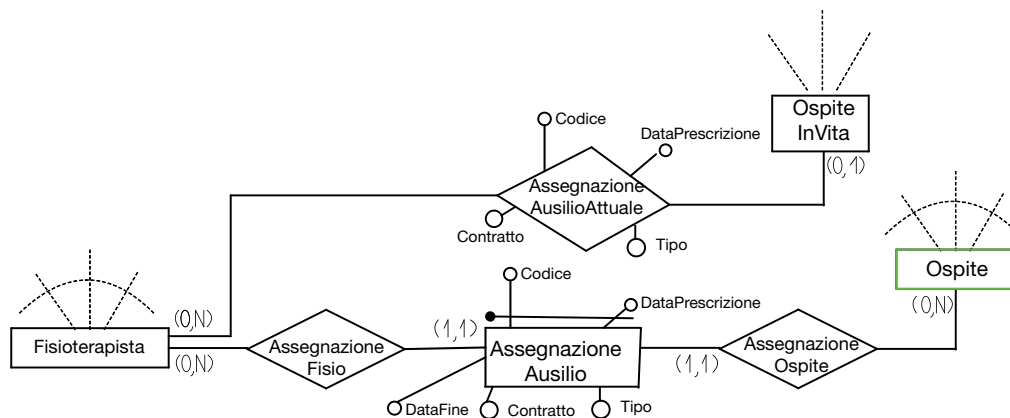


Figure 6: Schema concettuale Fisioterapisti

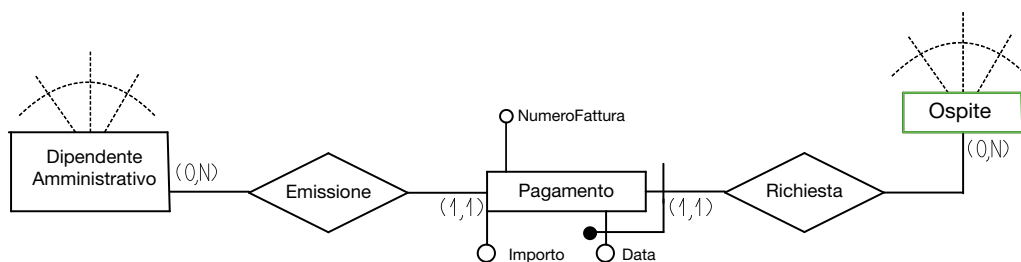


Figure 7: Schema concettuale Dipendenti Amministrativi

## 5 Progettazione logica

### 5.1 Ospiti

#### 5.1.1 Rimozione delle gerarchie

per entrambe le gerarchie, si opta per una scelta federalista. Nel caso dell'ospite, questa permette di non perdere i vincoli di references per cui solo un ospite in vita può avere un ricovero ed una dieta attuali e solo per un ospite deceduto possa essere registrata la data di decesso. Questo viene fatto tramite il vincolo 9, la creazione della tabella OSPITIDECEDUTI con attributo DataDecesso e la tabella OPSITIINVITA con attributi sulla dieta attuale.

Nel caso dei luoghi di appartenenza, la scelta federalista permette di accedere in maniera separata ai domicili privati e alle strutture di provenienza, e ciò risulta più in linea con le esigenze della struttura.

#### 5.1.2 Schema E-R ristrutturato

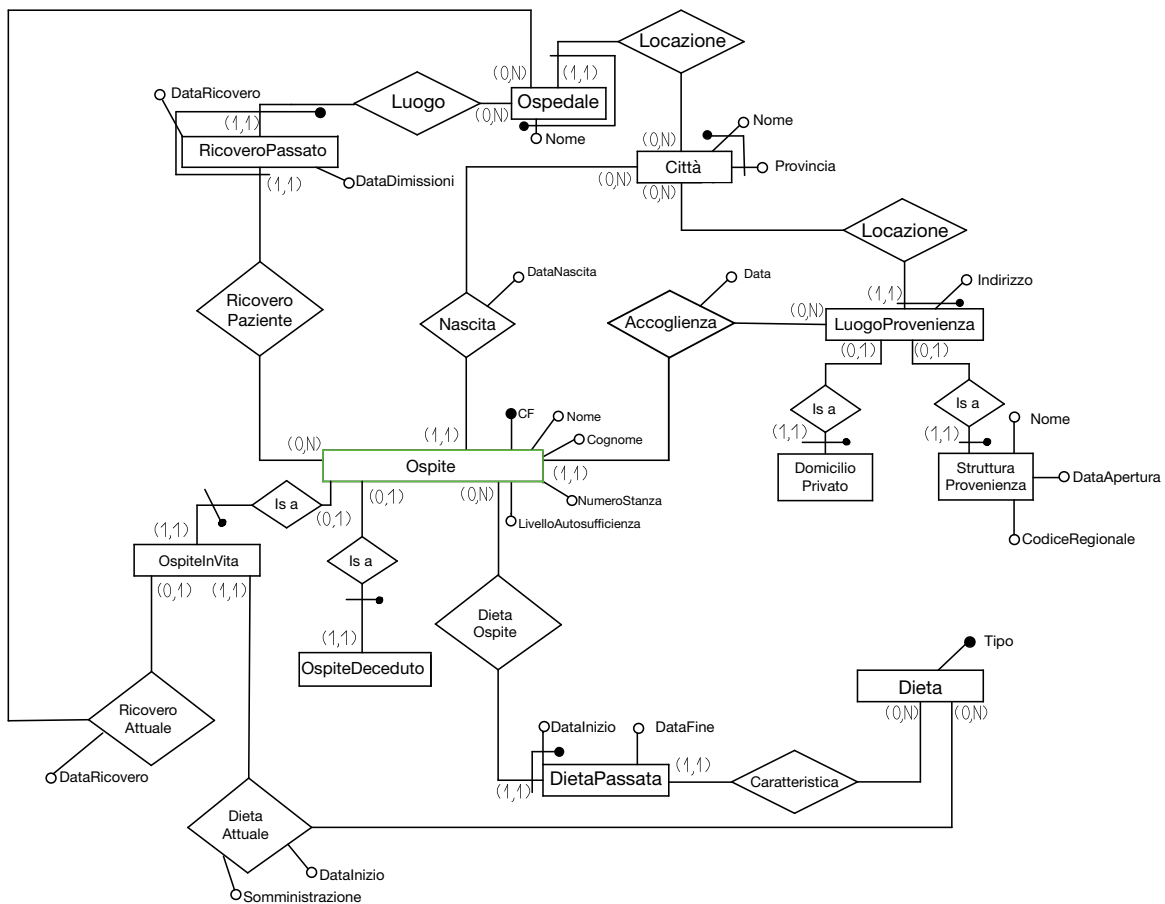


Figure 8: Schema concettuale Ospiti Ristrutturato

#### 5.1.3 Schema Logico

CITTA' (Nome, Provincia, Nazione)

LUOGHIPROVENIENZA (Indirizzo, CittàNome, CittàProvincia)

DOMICILIPRIVATIDIPROVENIENZA (Indirizzo, CittàNome, CittàProvincia)

STRUTTUREDIPROVENIENZA (Indirizzo, CittàNome, CittàProvincia, Nome, DataApertura, CodiceRegionale)

DIETE (Tipo)

OSPITI (CF, Nome, Cognome, NumeroStanza, LivelloAutosufficienza, CittàNascitaNome, CittàNascitaProvincia,

DataNascita, LuogoProvenienzaIndirizzo, LuogoProvenienzaCittàNome, LuogoProvenienzaCittàProvincia, DataAccoglienza)  
 OSPITIDECEDUTI (CF, DataDecesso)  
 OSPITIINVITA (CF, DietaAttuale, DataInizioDieta, Somministrazione)  
 OSPEDALI (Nome, CittàNome, CittàProvincia)  
 RICOVERIATTUALI (CFOspite, NomeOspedale, CittàOspedaleNome, CittàOspedaleProvincia, DataInizioRi-  
 covero)  
 RICOVERIPASSATI (CFOspite, NomeOspedale, CittàOspedaleNome, CittàOspedaleProvincia, DataInizioRicovero,  
 DataDimissioni)  
 DIETEPASSATE (CFOspite, TipoDieta, DataInizio, DataFine)

Con i vincoli:

1. LUOGHIPROVENIENZA (CittàNome, CittàProvincia)  $\subseteq$  CITTA' (Nome, Provincia)
2. DOMICILIPRIVATIDIPROVENIENZA (CittàNome, CittàProvincia)  $\subseteq$  CITTA' (Nome, Provincia)
3. STRUTTREDIPROVENIENZA (CittàNome, CittàProvincia)  $\subseteq$  CITTA' (Nome, Provincia)
4. OSPITI (CittàNascitaNome, CittàNascitaProvincia)  $\subseteq$  CITTA' (Nome, Provincia)
5. OSPITI (LuogoProvenienzaIndirizzo, LuogoProvenienzaCittàNome, LuogoProvenienzaCittàProvincia)  $\subseteq$  LUOGHIPROVENIENZA (Indirizzo, CittàNome, CittàProvincia)
6. OSPITIDECEDUTI (CF)  $\subseteq$  OSPITI (CF)
7. OSPITIINVITA (CF)  $\subseteq$  OSPITI (CF)
8. OSPITIINVITA (DietaAttuale)  $\subseteq$  DIETE (Tipo)
9. RICOVERIATTUALI (CFOspite)  $\subseteq$  OSPITIINVITA (CF)
10. RICOVERIATTUALI (NomeOspedale, CittàOspedaleNome, CittàOspedaleProvincia)  $\subseteq$  OSPEDALI (Nome, CittàNome, CittàProvincia)
11. RICOVERIPASSATI (CFOspite)  $\subseteq$  OSPITI (CF)
12. RICOVERIPASSATI (NomeOspedale, CittàOspedaleNome, , CittàOspedaleProvincia)  $\subseteq$  OSPEDALI (Nome, CittàNome, CittàProvincia)
13. DIETEPASSATE (CFOspite)  $\subseteq$  OSPITI (CF)
14. DIETEPASSATE (TipoDieta)  $\subseteq$  DIETE (Tipo)

## 5.2 Parenti

In questo caso non è necessario operare una ristrutturazione.

### 5.2.1 Schema Logico

PARENTI (CF, Nome, Cognome, DataNascita, NumeroDiTelefono, CittàResidenzaNome, CittàResidenzaProvincia)

PARENTELE (CFParente, CFOspite, Grado)

VISITE (CFParente, CFOspite, PortinaioControllo, Data, Orainizio)

Con i vincoli:

1. PARENTI (CittàResidenzaNome, CittàResidenzaProvincia)  $\subseteq$  CITTA' (Nome, Provincia)
2. PARNTELE(CFParente)  $\subseteq$  PARENTI (CF)
3. PARNTELE(CFOspite)  $\subseteq$  OSPITI (CF)
4. VISITE(CFParente)  $\subseteq$  PARENTI (CF)
5. VISITE(CFOspite)  $\subseteq$  OSPITI (CF)



## 5.3 Lavoratori

### 5.3.1 Rimozione delle gerarchie

Si opta per una scelta federalista in modo da avere accesso separato alle varie categorie di Lavoratori e di Dipendenti ma non rendere troppo complesse le numerose relazioni che hanno sia Lavoratore che Dipendente.

### 5.3.2 Schema E-R ristrutturato

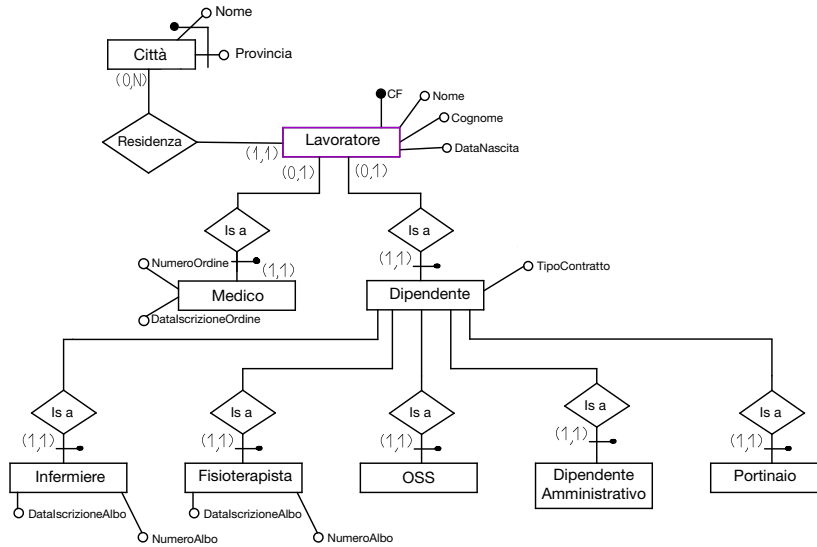


Figure 9: Schema concettuale Lavoratori Ristrutturato

### 5.3.3 Schema Logico

LAVORATORI (CF, Nome, Cognome, DataNascita, CittàResidenzaNome, CittàResidenzaProvincia)  
MEDICI (CF, NumeroOrdine, DataIscrizioneOrdine)  
DIPENDENTI (CF, TipoContratto)  
INFERMIERI (CF, NumeroAlbo, DataIscrizioneAlbo)  
FISIOTERPISTI (CF, NumeroAlbo, DataIscrizioneAlbo)  
OSS (CF)  
DIPENDENTIAMMINISTRATIVI (CF)  
PORTINAI (CF)

Con i vincoli:

1. LAVORATORI (CittàResidenzaNome, CittàResidenzaProvincia)  $\subseteq$  CITTA' (Nome, Provincia)
2. MEDICI (CF)  $\subseteq$  LAVORATORI (CF)
3. DIPENDENTI (CF)  $\subseteq$  LAVORATORI (CF)
4. INFERMIERI (CF)  $\subseteq$  DIPENDENTI (CF)
5. FISIOTERPISTI (CF)  $\subseteq$  DIPENDENTI (CF)
6. OSS (CF)  $\subseteq$  DIPENDENTI (CF)
7. DIPENDENTIAMMINISTRATIVI (CF)  $\subseteq$  DIPENDENTI (CF)
8. PORTINAI (CF)  $\subseteq$  DIPENDENTI (CF)

## 5.4 Medici

In questo caso non è necessario operare una ristrutturazione.

### 5.4.1 Schema Logico

RAPPORTIMEDICI (Medico, Ospite)

VISITEMEDICHE (Medico, Ospite, Data)

DIAGNOSI (NumeroReferto, PartedelCorpo, Medico, Ospite, DataVisita)

PRESCRIZIONIFARMACI (NumeroReferto, CodiceFarmaco, Quantitativo, Cadenza, Tipo)

DOSIVACCINOOSPITI (Medico, Ospite, NumeroDose, Data)

DOSIVACCINOLAVORATORI (Medico, Lavoratore, NumeroDose, Data)

Con i vincoli:

1. RAPPORTIMEDICI (Medico)  $\subseteq$  MEDICI (CF)
2. RAPPORTIMEDICI (Ospite)  $\subseteq$  OSPITI(CF)
3. VISITE (Medico,Ospite)  $\subseteq$  RAPPORTIMEDICI (Medico, Ospite)
4. DIAGNOSI (Medico,Ospite,DataVisita)  $\subseteq$  VISITE (Medico,Ospite,Data)
5. PRESCRIZIONIFARMACI (NumeroReferto)  $\subseteq$  DIAGNOSI (NumeroReferto)
6. DOSIVACCINOOSPITI (Medico)  $\subseteq$  MEDICI (CF)
7. DOSIVACCINOOSPITI (Ospite)  $\subseteq$  OSPITI(CF)
8. DOSIVACCINOLAVORATORI (Medico)  $\subseteq$  MEDICI (CF)
9. DOSIVACCINOLAVORATORI (Ospite)  $\subseteq$  LAVORATORI(CF)

Si può ora comprendere il senso della creazione dell'entità RapportoMedico. Essa permette la creazione del vincolo di references numero 3, secondo cui la visita avviene solo tra medici ed ospiti che condividono un rapporto medico, e a cascata la diagnosi avviene solo se c'è la visita (vincolo 4) e la prescrizione solo se c'è la diagnosi (vincolo 5).

## 5.5 Infermieri ed OSS

### 5.5.1 Rimozione delle gerarchie

In questo caso si è optato per una scelta centralista andando ad accorpare entrambe le gerarchie nell'entità madre, aggiungendo un attributo 'Tipo' e aggiungendo gli opportuni attributi opzionali. Si noti che l'attributo 'Tipo' in Prestazione risulterà ridondante (poiché questo deve coincidere con quello di mansione), ma può essere utile per un ulteriore controllo.

### 5.5.2 Schema E-R ristrutturato

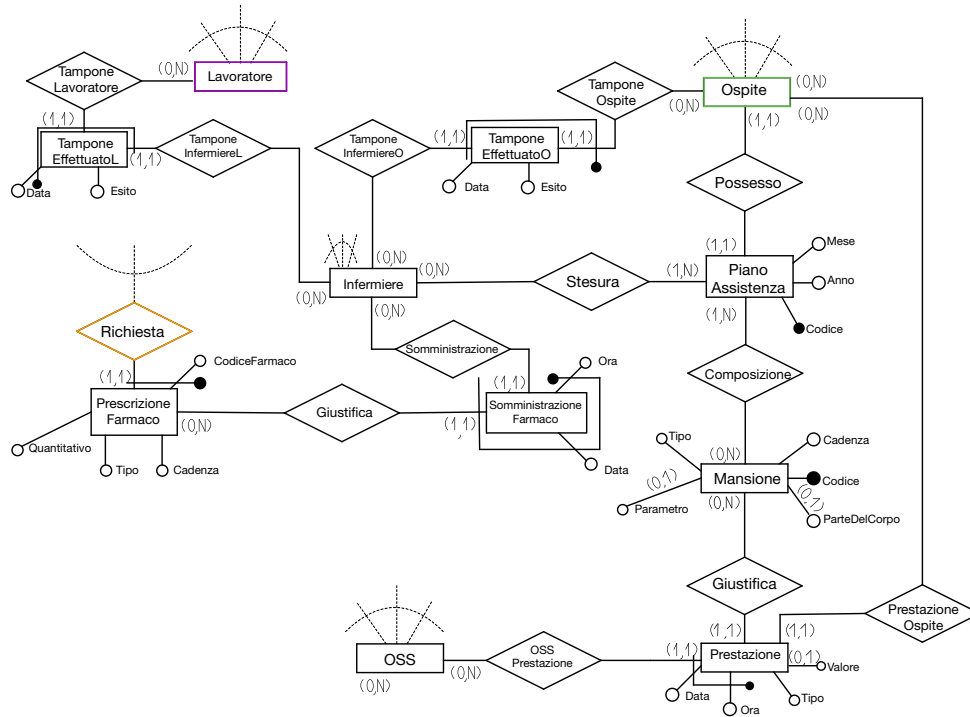


Figure 10: Schema concettuale Infermieri ed OSS Ristrutturato

### 5.5.3 Schema Logico

SOMMINISTRAZIONIFARMACI (NumeroReferto, CodiceFarmaco, Infermiere, Data, Ora)  
TAMPONIEFFETTUALAVORATORI (Infermiere, Lavoratore, Data, Esito)  
TAMPONIEFFETTUALOSPITI (Infermiere, Ospite, Data, Esito)  
PIANIASSISTENZA (Codice, Mese, Anno, Ospite)  
STESUREPIANI (Infermiere, CodicePiano)  
MANISONI (Codice, Cadenza, Tipo, ParteDelCorpo\*, Parametro\*)  
COMPOSIZIONEPIANI (Piano, Mansione)  
PRESTAZIONI (CodiceMansione, Oss, Ospite, Data, Ora, Tipo, Valore\*)

Con i vincoli:

1. SOMMINISTRAZIONIFARMACI (NumeroReferto, CodiceFarmaco)  $\subseteq$  PRESCRIZIONIFARMACI (NumeroReferto, CodiceFarmaco)
2. TAMPONIEFFETTUALAVORATORI (Infermiere)  $\subseteq$  INFERMIERI (CF)
3. TAMPONIEFFETTUALAVORATORI (Lavoratore)  $\subseteq$  LAVORATORI (CF)

4. TAMPONIEFFETTUAOSPITI (Infermiere)  $\subseteq$  INFERMIERI (CF)
5. TAMPONIEFFETTUAOSPITI (Ospite)  $\subseteq$  OSPITI(CF)
6. PIANIASSISTENZA (Ospite)  $\subseteq$  OSPITI(CF)
7. STESUREPIANI (Infermiere)  $\subseteq$  INFERMIERI (CF)
8. STESUREPIANI (CodicePiano)  $\subseteq$  PIANIASSISTENZA (Codice)
9. COMPOSIZIONEPIANI (Piano)  $\subseteq$  PIANIASSISTENZA (Codice)
10. COMPOSIZIONEPIANI (Mansione)  $\subseteq$  MANISONI (Codice)
11. PRESTAZIONI (CodiceMansione)  $\subseteq$  MANISONI (Codice)
12. PRESTAZIONI (Oss)  $\subseteq$  OSS(CF)
13. PRESTAZIONI (Ospite)  $\subseteq$  OSPITI(CF)

## 5.6 Fisioterapisti e Dipendenti amministrativi

Non è necessario operare una ristrutturazione.

### 5.6.1 Schema Logico

STORICOASSEGNAZIONEAUSILI (Codice, Contratto, DataAssegnazione, Tipo, Fisioterapista, Ospite, DataFine)

ASSEGNAZIONEAUSILIATTUALI (Codice, Contratto, DataAssegnazione, Tipo, Fisioterapista, Ospite)

Con i vincoli:

1. STORICOASSEGNAZIONEAUSILI (Fisioterapista)  $\subseteq$  FISIOTERAPISTI(CF)
2. STORICOASSEGNAZIONEAUSILI (Ospite)  $\subseteq$  OSPITI(CF)
3. ASSEGNAZIONEAUSILIATTUALI (Fisioterapista)  $\subseteq$  FISIOTERAPISTI(CF)
4. ASSEGNAZIONEAUSILIATTUALI (Ospite)  $\subseteq$  OSPITIINVITA(CF)

PAGAMENTI (Data, Ospite, NumeroFattura, Importo, DipendenteAmministrativo)

Con i vincoli:

1. PAGAMENTI (Ospite)  $\subseteq$  OSPITI(CF)
2. PAGAMENTI (DipendenteAmministrativo)  $\subseteq$  DIPEDENTIAMMINISTRATIVI (CF)

## 6 Implementazione SQL

### 6.1 Creazione e popolamento delle tabelle

#### 6.1.1 Ospiti

```
1  create table Citta
2  (
3  Nome varchar(40),
4  Provincia varchar(40),
5  Nazione varchar(40),
6  primary key (Nome , Provincia)
7  );
8
9  create table LuoghiProvenienza
10 (
11 Indirizzo varchar(40) ,
12 CittaNome varchar(40),
13 CittaProvincia varchar(40),
14 primary key (Indirizzo,CittaNome, CittaProvincia),
15 foreign key (CittaNome, CittaProvincia) references Citta (Nome, Provincia)
16 );
17
18 create table DomiciliPrivatiDiProvenienza
19 (Indirizzo varchar(40),
20 CittaNome varchar(40),
21 CittaProvincia varchar(40),
22 primary key (Indirizzo,CittaNome, CittaProvincia),
23 foreign key (CittaNome, CittaProvincia) references Citta (Nome, Provincia)
24 );
25
26 create table StruttureDiProvenienza
27 (Indirizzo varchar(40),
28 CittaNome varchar(40),
29 CittaProvincia varchar(40),
30 Nome varchar(40),
31 DataApertura date,
32 CodiceRegionale char(4),
33 primary key (Indirizzo,CittaNome, CittaProvincia),
34 foreign key (CittaNome, CittaProvincia) references Citta (Nome, Provincia)
35 );
36
37 create table Diete
38 (Tipo varchar(40) primary key
39 );
40
41 create table Ospiti
42 (CF char(16) primary key,
43 Nome varchar(40),
44 Cognome varchar(40),
45 NumeroStanza int,
46 LivelloAutosufficienza int,
47 CittaNascitaNome varchar(40),
48 CittaNascitaProvincia varchar(40),
49 DataNascita date,
50 DataAccoglienza date,
51 LuogoProvenienzaIndirizzo varchar(40),
52 LuogoProvenienzaCittaNome varchar(40),
53 LuogoProvenienzaCittaProvincia varchar(40),
54 foreign key (CittaNascitaNome, CittaNascitaProvincia) references Citta (Nome, Provincia),
55 foreign key (LuogoProvenienzaIndirizzo, LuogoProvenienzaCittaNome, LuogoProvenienzaCittaProvincia) references
56 LuoghiProvenienza (Indirizzo, CittaNome, CittaProvincia)
57 );
58
59 create table OspitiDeceduti
60 (CF char(16) primary key references Ospiti(CF),
61 DataDecesso date
```

```

62 );
63
64 create table OspitiInVita
65 (CF char(16) primary key references Ospiti(CF),
66 DietaAttuale varchar(40) references Diete(Tipo),
67 DataInizioDieta date,
68 Somministrazione varchar(30)
69 );
70
71 create table Ospedali
72 (Nome varchar (60),
73 CittaNome varchar(40),
74 CittaProvincia varchar(40),
75 primary key (Nome, CittaNome, CittaProvincia),
76 foreign key (CittaNome, CittaProvincia) references Citta (Nome, Provincia)
77 );
78
79 create table RicoveriAttuali
80 (
81 CFospite char (16) references OspitiInVita (CF),
82 NomeOspedale varchar (60),
83 CittaOspedaleNome varchar(40),
84 CittaOspedaleProvincia varchar(40),
85 DataInizioRicovero date,
86 primary key (CFospite, NomeOspedale, CittaOspedaleNome, CittaOspedaleProvincia),
87 foreign key (NomeOspedale, CittaOspedaleNome, CittaOspedaleProvincia) references Ospedali (Nome,CittaNome,CittaProvincia)
88 );
89
90 create table RicoveriPassati
91 (CFospite char (16) references Ospiti(CF),
92 NomeOspedale varchar (60),
93 CittaOspedaleNome varchar(40),
94 CittaOspedaleProvincia varchar(40),
95 DataInizioRicovero date,
96 DataDimissioni date,
97 primary key (CFospite, NomeOspedale, CittaOspedaleNome, CittaOspedaleProvincia,DataInizioRicovero ),
98 foreign key (NomeOspedale, CittaOspedaleNome, CittaOspedaleProvincia) references Ospedali (Nome,CittaNome,CittaProvincia)
99 );
100
101 create table DietePassate
102 (CFospite char (16) references Ospiti(CF),
103 TipoDieta varchar(40),
104 DataInizio date,
105 DataFine date,
106 primary key (CFospite, TipoDieta, DataInizio ),
107 foreign key (TipoDieta) references Diete(Tipo)
108 );
109
110
111 insert into Citta values
112 ('Trieste','Trieste','Italia'),
113 ('Este','Padova','Italia'),
114 ('Codroipo','Udine','Italia');
115
116 insert into LuoghiProvenienza values
117 ('Via Foralnini 131','Trieste','Trieste'),
118 ('Via Carducci 42','Este','Padova'),
119 ('Via Marchesetti 37', 'Trieste','Trieste')
120 ;
121
122 insert into DomiciliPrivatiDiProvenienza values
123 ('Via Foralnini 131', 'Trieste','Trieste'),
124 ('Via Carducci 42','Este','Padova');
125
126 insert into StruttureDiProvenienza values
127 ('Via Marchesetti 37', 'Trieste','Trieste', 'Casa Serena', '01-MAY-1987', '0001');
128
129 insert into Diete values

```

```

130 ('Regolare'),
131 ('Ipoglicidica'),
132 ('Iposodica'),
133 ('Ipoproteica'),
134 ('Vegetariana');
135
136 insert into Ospiti values
137 ('RVSNNK90B07I122X','Ettore','Siciliano',34,1,'Trieste','Trieste','10-MAY-1950', '12-MAY-2020','Via Foralnini 131',
138 'Trieste','Trieste'),
139 ('DXLSPM52A02G600G','Arsenio','Lo Duca',21,2,'Este','Padova','7-FEB-1942', '13-MAY-2022','Via Carducci 42','Este','Padova'),
140 ('ZBTFQF51L65F174I','Dorotea','Lombardi',12,3,'Codroipo','Udine','8-JUL-1932', '14-JUN-2022','Via Marchesetti 37',
141 'Trieste','Trieste'),
142 ('JCVHSS97E48B744F','Ettore','Cociani',34,3,'Trieste','Trieste','1-MAY-1950', '8-MAY-2005','Via Foralnini 131',
143 'Trieste','Trieste');
144
145 insert into OspitiDeceduti values
146 ('RVSNNK90B07I122X','5-MAY-2022'),
147 ('JCVHSS97E48B744F','7-MAY-2022');
148
149 insert into OspitiInVita values
150 ('DXLSPM52A02G600G','Regolare','29-JUN-2022','Frullata'),
151 ('ZBTFQF51L65F174I','Vegetariana','24-JUN-2022','Intera');
152
153 insert into Ospedali values
154 ('Ospedale di Cattinara','Trieste','Trieste'),
155 ('Ospedale Maggiore','Trieste','Trieste');
156
157 insert into RicoveriAttuali values
158 ('DXLSPM52A02G600G','Ospedale di Cattinara','Trieste','Trieste','7-DEC-2022');
159
160 insert into RicoveriPassati values
161 ('DXLSPM52A02G600G','Ospedale Maggiore','Trieste','Trieste','7-DEC-2002','7-FEB-2003'),
162 ('RVSNNK90B07I122X','Ospedale di Cattinara','Trieste','Trieste','7-NOV-2005','3-FEB-2006');
163
164 insert into DietePassate values
165 ('RVSNNK90B07I122X','Ipoproteica','10-MAY-2021','10-OCT-2021'),
166 ('DXLSPM52A02G600G','Iposodica','7-FEB-2004','23-OCT-2021'),
167 ('ZBTFQF51L65F174I','Regolare','17-MAY-2004','23-JUN-2020');

```

## 6.1.2 Parenti

```

1  create table Parenti
2  (CF char(16) primary key,
3   Nome varchar(40),
4   Cognome varchar(40),
5   DataNascita date,
6   NumeroDiTelefono varchar (20),
7   CittaResidenzaNome varchar(40),
8   CittaResidenzaProvincia varchar(40),
9   foreign key (CittaResidenzaNome, CittaResidenzaProvincia) references Citta (Nome, Provincia)
10  );
11
12  create table Parentele
13  (CF0spite char(16),
14   CFParente char(16),
15   Grado varchar(30),
16   primary key (CF0spite,CFParente)
17  );
18
19  create table Visite
20  (CFParente char(16) references Parenti (CF),
21   CF0spite char(16) references Ospiti (CF),
22   PortinaioControllo char(16),
23   Data date,
24   Orainizio time,
25   primary key (CFParente,CF0spite,Data)

```



```

26 );
27
28
29 insert into Parenti values
30 ('FLNYDF98T68A825A', 'Rosalia', 'Marcelo', '19-MAR-2002', '0351 1278023', 'Trieste', 'Trieste'),
31 ('YLTFH56C25L048S', 'Manuela', 'Bruno', '26-OCT-1982', '0358 3849175', 'Trieste', 'Trieste'),
32 ('VLBXVR31E50G041B', 'Lidia', 'Gallo', '16-JUN-1982', '0324 5871722', 'Trieste', 'Trieste'),
33 ('SFQTB62E63C879M', 'Artemisia', 'Trentini', '16-DEC-1999', '0317 7605808', 'Este', 'Padova'),
34 ('ZBLTVA42S51B262Y', 'Alcide', 'Esposito', '19-MAR-2002', '0340 2327004', 'Este', 'Padova'),
35 ('ZLCFZF71L49L342Q', 'Ubaldo', 'Esposito', '26-OCT-1982', '0318 3531293', 'Este', 'Padova');
36
37 insert into Parentele values
38 ('RVSNNV90B07I122X', 'FLNYDF98T68A825A', 'nipote'),
39 ('RVSNNV90B07I122X', 'YLTFH56C25L048S', 'figlia'),
40 ('DXLSPM52A02G600G', 'VLBXVR31E50G041B', 'nipote'),
41 ('DXLSPM52A02G600G', 'SFQTB62E63C879M', 'figlia'),
42 ('ZBTFQF51L65F174I', 'ZBLTVA42S51B262Y', 'genere'),
43 ('ZBTFQF51L65F174I', 'ZLCFZF71L49L342Q', 'genere');
44
45 insert into Visite values
46 ('VLBXVR31E50G041B', 'DXLSPM52A02G600G', 'TSDRMR81M31F441U', '13-JAN-2023', '13:15:12'),
47 ('VLBXVR31E50G041B', 'DXLSPM52A02G600G', 'TSDRMR81M31F441U', '07-DEC-2021', '17:15:12'),
48 ('VLBXVR31E50G041B', 'DXLSPM52A02G600G', 'TSDRMR81M31F441U', '11-JAN-2023', '13:15:11'),
49 ('SFQTB62E63C879M', 'DXLSPM52A02G600G', 'THBRXP67S49G717W', '12-JAN-2023', '07:15:12'),
50 ('SFQTB62E63C879M', 'DXLSPM52A02G600G', 'THBRXP67S49G717W', '13-DEC-2022', '09:15:44'),
51 ('ZLCFZF71L49L342Q', 'ZBTFQF51L65F174I', 'THBRXP67S49G717W', '13-JAN-2022', '10:15:12');

```

### 6.1.3 Laboratori

```

1 create table Laboratori
2 (CF char(16) primary key,
3 Nome varchar(40),
4 Cognome varchar(40),
5 DataNascita date,
6 CittaResidenzaNome varchar(40),
7 CittaResidenzaProvincia varchar(40),
8 foreign key (CittaResidenzaNome, CittaResidenzaProvincia) references Citta (Nome, Provincia)
9 );
10
11 create table Medici
12 (CF char(16) primary key references Laboratori(CF),
13 NumeroOrdine char(4),
14 DataIscrizioneOrdine date
15 );
16
17 create table Dipendenti
18 (CF char(16) primary key references Laboratori(CF),
19 TipoContratto varchar(40)
20 );
21
22 create table Infermieri
23 (CF char(16) primary key references Dipendenti(CF),
24 NumeroAlbo char(4),
25 DataIscrizioneAlbo date
26 );
27
28 create table Fisioterapisti
29 (CF char(16) primary key references Dipendenti(CF),
30 NumeroAlbo char(4),
31 DataIscrizioneAlbo date
32 );
33
34 create table OSS
35 (CF char(16) primary key references Dipendenti(CF)
36 );
37

```

```

38 create table DipendentiAmministrativi
39 (CF char(16) primary key references Dipendenti(CF)
40 );
41
42 create table Portinai
43 (CF char(16) primary key references Dipendenti(CF)
44 );
45
46
47 insert into Lavoratori values
48 ('XRZQWQ36A26F680X', 'Dimitri', 'Conti', '5-12-1987', 'Trieste', 'Trieste'),
49 ('SFCFR52L06E952I', 'Clizia', 'Padovano', '29-MAY-1952', 'Trieste', 'Trieste'),
50 ('JEOPHG66H51L809M', 'Viviano', 'Panicucci', '7-JUL-1984', 'Trieste', 'Trieste'),
51 ('PDDTCZ54A50D043X', 'Adamo', 'Costa', '8-NOV-1988', 'Trieste', 'Trieste'),
52 ('GPACGL62D08E336B', 'Alcide', 'Lo Duca', '24-OCT-2001', 'Trieste', 'Trieste'),
53 ('NGVHVS83R49F016S', 'Cosimo', 'Pisano', '12-JUL-1993', 'Este', 'Padova'),
54 ('VSSRDC79M15L347H', 'Amedeo', 'De Luca', '18-MAY-1956', 'Este', 'Padova'),
55 ('PGZPZB90R65B672R', 'Silvia', 'Lori', '1-JUN-1997', 'Este', 'Padova'),
56 ('NBRGKJ37D18A704J', 'Giacchino', 'Iadanza', '22-MAY-1978', 'Este', 'Padova'),
57 ('YVGZMD57D42I164J', 'Maria Pia', 'Marino', '10-JUL-1953', 'Codroipo', 'Udine'),
58 ('XXJFZV44T62E372X', 'Felice', 'Bianchi', '21-SEP-1947', 'Codroipo', 'Udine'),
59 ('SBFPHF39S25B437G', 'Isidora', 'Russo', '11-APR-1988', 'Codroipo', 'Udine'),
60 ('VFFVWR70T47C629Y', 'Livia', 'Napolitani', '17-APR-1987', 'Codroipo', 'Udine');
61
62
63 insert into Medici values
64 ('XRZQWQ36A26F680X', '4329', '5-DEC-1999'),
65 ('SFCFR52L06E952I', '4738', '21-MAY-1995');
66
67
68 insert into Dipendenti values
69 ('JEOPHG66H51L809M', 'Determinato'),
70 ('PDDTCZ54A50D043X', 'Indeterminato'),
71 ('GPACGL62D08E336B', 'Determinato'),
72 ('NGVHVS83R49F016S', 'Determinato'),
73 ('VSSRDC79M15L347H', 'Indeterminato'),
74 ('PGZPZB90R65B672R', 'Indeterminato'),
75 ('NBRGKJ37D18A704J', 'Indeterminato'),
76 ('YVGZMD57D42I164J', 'Indeterminato'),
77 ('XXJFZV44T62E372X', 'Indeterminato'),
78 ('SBFPHF39S25B437G', 'Indeterminato'),
79 ('VFFVWR70T47C629Y', 'Indeterminato')
80 );
81
82
83 insert into Infermieri values
84 ('JEOPHG66H51L809M', '4329', '21-MAY-1995'),
85 ('PDDTCZ54A50D043X', '4738', '5-MAY-1998');
86
87
88 insert into Fisioterapisti values
89 ('GPACGL62D08E336B', '3214', '21-MAY-2005'),
90 ('NGVHVS83R49F016S', '7689', '21-MAY-2001');
91
92 insert into OSS values
93 ('VSSRDC79M15L347H'),
94 ('PGZPZB90R65B672R');
95
96 insert into DipendentiAmministrativi values
97 ('NBRGKJ37D18A704J'),
98 ('YVGZMD57D42I164J');
99
100 insert into Portinai values
101 ('XXJFZV44T62E372X'),
102 ('SBFPHF39S25B437G'),
103 ('VFFVWR70T47C629Y')
104 );

```

#### 6.1.4 Medici

```
1  create table RapportiMedici
2  (Medico char(16) references Medici(CF),
3   Ospite char(16) references Ospiti(CF),
4   primary key (Medico,Ospite)
5  );
6
7  create table VisiteMediche
8  (Medico char(16) references Medici(CF),
9   Ospite char(16) references Ospiti(CF),
10  Data date,
11  foreign key (Medico,Ospite) references RapportiMedici (Medico,Ospite)
12  primary key (Medico,Ospite,Data)
13  );
14
15  create table Diagnosi
16  (NumeroReferto char(4) primary key,
17   ParteDelCorpo varchar(30),
18   Medico char(16),
19   Ospite char(16),
20   DataVisita date,
21   foreign key (Medico,Ospite,DataVisita) references VisiteMediche(Medico,Ospite,Data)
22  );
23
24  create table PrescrizioniFarmaci
25  (NumeroReferto char(4) references Diagnosi(NumeroReferto),
26   CodiceFarmaco char(6),
27   Quantitativo varchar(30),
28   Cadenza varchar(30),
29   Tipo varchar(30),
30   Attiva bool
31  primary key(NumeroReferto,CodiceFarmaco)
32  );
33
34  create table DosiVaccinoOspiti
35  (Medico char(16) references Medici(CF),
36   Ospite char(16) references Ospiti(CF),
37   NumeroDose int,
38   Data date,
39  primary key (Medico, Ospite, NumeroDose)
40  );
41
42  create table DosiVaccinoLavoratori
43  (Medico char(16) references Medici(CF),
44   Lavoratore char(16) references Lavoratori(CF),
45   NumeroDose int,
46   Data date,
47  primary key (Medico, Lavoratore, NumeroDose)
48  );
49
50
51  insert into RapportiMedici values
52  ('XRZQWQ36A26F680X', 'RVSNVK90B07I122X'),
53  ('SFCRFR52L06E952I', 'DXLSPM52A02G600G'),
54  ('SFCRFR52L06E952I','ZBTFQF51L65F174I'
55  );
56
57  insert into VisiteMediche values
58  ('XRZQWQ36A26F680X', 'RVSNVK90B07I122X','19-MAR-2022'),
59  ('XRZQWQ36A26F680X', 'RVSNVK90B07I122X','26-DEC-2022'),
60  ('XRZQWQ36A26F680X', 'RVSNVK90B07I122X','19-JAN-2022'),
61  ('SFCRFR52L06E952I', 'DXLSPM52A02G600G','16-JAN-2023'),
62  ('SFCRFR52L06E952I', 'DXLSPM52A02G600G','17-APR-2021'),
63  ('SFCRFR52L06E952I','ZBTFQF51L65F174I','19-MAR-2022'
64  );
65
66  insert into Diagnosi values
```

```

67 ('1010','bracciodx','XRZQWQ36A26F680X','RVSNVK90B07I122X','19-MAR-2022'),
68 ('1011','bracciosx','XRZQWQ36A26F680X','RVSNVK90B07I122X','19-MAR-2022'),
69 ('1012','gambadx','XRZQWQ36A26F680X','RVSNVK90B07I122X','19-MAR-2022'),
70 ('1013','gambasx','SFCRFR52L06E952I','DXLSPM52A02G600G','16-JAN-2023'),
71 ('1014','torace','SFCRFR52L06E952I','DXLSPM52A02G600G','17-APR-2021'),
72 ('1015','piededx','SFCRFR52L06E952I','ZBTFQF51L65F174I','19-MAR-2022'
73 );
74
75 insert into PrescrizioniFarmaci values
76 ('1010','3456','3 gr','6 ore','fisso',true),
77 ('1010','5678','6 gr','3 ore','condizionato',true),
78 ('1010','1234','7 gr','2 ore','fisso',true),
79 ('1013','6789','10 gr','9 ore','condizionato',true),
80 ('1014','3256','3 gr','10 ore','fisso',true);
81
82 insert into DosiVaccinoOspiti values
83 ('XRZQWQ36A26F680X','RVSNVK90B07I122X',1,'17-MAY-2022'),
84 ('XRZQWQ36A26F680X','RVSNVK90B07I122X',2,'17-JUN-2022'),
85 ('XRZQWQ36A26F680X','RVSNVK90B07I122X',3,'17-DEC-2022'),
86 ('SFCRFR52L06E952I','DXLSPM52A02G600G',1,'7-MAY-2022'),
87 ('SFCRFR52L06E952I','DXLSPM52A02G600G',2,'17-JUN-2022'),
88 ('SFCRFR52L06E952I','DXLSPM52A02G600G',3,'3-DEC-2022');
89
90 insert into DosiVaccinoLavoratori values
91 ('XRZQWQ36A26F680X','XRZQWQ36A26F680X',1,'17-MAY-2022'),
92 ('XRZQWQ36A26F680X','SFCRFR52L06E952I',2,'17-JUN-2022'),
93 ('XRZQWQ36A26F680X','JEOPHG66H51L809M',3,'17-DEC-2022'),
94 ('SFCRFR52L06E952I','PDDTCZ54A50D043X',1,'7-MAY-2022'),
95 ('SFCRFR52L06E952I','PDDTCZ54A50D043X',2,'17-JUN-2022'),
96 ('SFCRFR52L06E952I','PDDTCZ54A50D043X',3,'3-DEC-2022'),
97 ('SFCRFR52L06E952I','YVGZMD57D42I164J',2,'3-DEC-2022');

```

### 6.1.5 Infermieri e OSS

```

1  create table SomministrazioneFarmaci
2  (
3  NumeroReferto char(4),
4  CodiceFarmaco char (4),
5  Infermiere char (16) references Infermieri (CF),
6  Data date,
7  Ora time,
8  foreign key (NumeroReferto,CodiceFarmaco) references PrescrizioniFarmaci (NumeroReferto,CodiceFarmaco),
9  primary key (Infermiere,Data,Ora)
10 );
11
12 create table TamponiEffettuatiLavoratori
13 (
14 Infermiere char (16) references Infermieri (CF),
15 Lavoratore char (16) references Lavoratori (CF),
16 Data date,
17 Esito varchar(20),
18 primary key (Infermiere,Lavoratore,Data)
19 );
20
21 create table TamponiEffettuatiOspiti
22 (
23 Infermiere char (16) references Infermieri (CF),
24 Ospite char (16) references Ospiti (CF),
25 Data date,
26 Esito varchar(20),
27 primary key (Infermiere,Ospite,Data)
28 );
29
30 create table PianiAssistenza
31 (
32 Codice char (4) primary key ,

```

```

33 Mese char(20),
34 Anno int,
35 Ospite char (16) references Ospiti (CF)
36 );
37
38 create table StesurePiani
39 (
40 Infermiere char (16) references Infermieri (CF) ,
41 CodicePiano char (4) references PianiAssistenza (Codice),
42 primary key (Infermiere,CodicePiano)
43 );
44
45 create table Mansioni
46 (
47 Codice char (4) primary key,
48 Cadenza varchar(30),
49 Tipo varchar(30),
50 ParteDelCorpo varchar(30),
51 Parametro varchar(30)
52 );
53
54 create table ComposizionePiani
55 (
56 CodicePiano char (4) references PianiAssistenza (Codice) ,
57 CodiceMansione char (4) references Mansioni (Codice),
58 primary key (CodicePiano,CodiceMansione)
59 );
60
61 create table Prestazioni
62 (
63 CodiceMansione char (4) references Mansioni (Codice),
64 OSS char (16) references OSS (CF),
65 Ospite char (16) references Ospiti (CF),
66 Data date,
67 Ora time,
68 Tipo varchar(30),
69 Valore varchar(30),
70 primary key (OSS,Data,Ora)
71 );
72
73
74 insert into SomministrazioneFarmaci values
75 ('1010','3456','JEOPHG66H51L809M','13-JAN-2023','17:10:22'),
76 ('1010','3456','JEOPHG66H51L809M','14-JAN-2023','18:10:22'),
77 ('1010','3456','JEOPHG66H51L809M','13-JAN-2023','17:01:22'),
78 ('1010','5678','PDDTCZ54A50D043X','15-JAN-2023','4:10:25'),
79 ('1014','3256','PDDTCZ54A50D043X','13-JAN-2023','7:10:22');
80
81 insert into TamponiEffettuatiLavoratori values
82 ('JEOPHG66H51L809M','XRZQWQ36A26F680X','14-JAN-2023', false),
83 ('JEOPHG66H51L809M','SFCRFR52L06E952I','14-JAN-2023', false),
84 ('JEOPHG66H51L809M','JEOPHG66H51L809M','15-JAN-2023', false);
85
86 insert into TamponiEffettuatiOspiti values
87 ('JEOPHG66H51L809M','DXLSPM52A02G600G','14-DEC-2022', false),
88 ('JEOPHG66H51L809M','DXLSPM52A02G600G','14-JAN-2023', true),
89 ('JEOPHG66H51L809M','ZBTFQF51L65F174I','7-DEC-2022', false),
90 ('JEOPHG66H51L809M','ZBTFQF51L65F174I','15-JAN-2023', false);
91
92 insert into PianiAssistenza values
93 ('2345','Gennaio','2023','DXLSPM52A02G600G'),
94 ('2346','Dicembre','2022','DXLSPM52A02G600G'),
95 ('2347','Dicembre','2022','ZBTFQF51L65F174I'),
96 ('2348','Gennaio','2023','ZBTFQF51L65F174I');
97
98 insert into StesurePiani values
99 ('JEOPHG66H51L809M','2345'),
100 ('JEOPHG66H51L809M','2346'),

```

```

101 ('JEOPHG66H51L809M','2347'),
102 ('JEOPHG66H51L809M','2348');
103
104 insert into Mansioni values
105 ('0001','giornaliera','misurazione',null,'pressione'),
106 ('0002','giornaliera','medicazione','piede dx',null),
107 ('0003','giornaliera','medicazione','piede sx',null),
108 ('0004','giornaliera','medicazione','mano dx',null),
109 ('0005','settimanale','medicazione','mano dx',null);
110
111 insert into ComposizionePiani values
112 ('2345','0001'),
113 ('2345','0002'),
114 ('2346','0001'),
115 ('2347','0001'),
116 ('2348','0002'),
117 ('2345','0005'),
118 ('2348','0005');
119
120 insert into Prestazioni values
121 ('0001','VSSRDC79M15L347H','DXLSPM52A02G600G','13-JAN-2023','7:10:22','misurazione','97'),
122 ('0002','VSSRDC79M15L347H','DXLSPM52A02G600G','13-JAN-2023','17:10:22','medicazione',null),
123 ('0002','VSSRDC79M15L347H','DXLSPM52A02G600G','14-JAN-2023','7:10:22','medicazione',null),
124 ('0002','VSSRDC79M15L347H','ZBTFQF51L65F174I','14-JAN-2023','7:45:22','medicazione',null),
125 ('0002','PGZPZB90R65B672R','ZBTFQF51L65F174I','13-JAN-2023','7:10:22','medicazione',null),
126 ('0002','PGZPZB90R65B672R','ZBTFQF51L65F174I','12-JAN-2023','6:10:22','medicazione',null),
127 ('0005','PGZPZB90R65B672R','DXLSPM52A02G600G','16-DEC-2022','6:10:22','medicazione',null),
128 ('0005','PGZPZB90R65B672R','ZBTFQF51L65F174I','12-DEC-2022','6:10:22','medicazione',null);

```

### 6.1.6 Fisioterapisti e Dipendenti Amministrativi

```

1 create table StoricoAssegnazioneAusili
2 (
3     Codice char (4),
4     Contratto varchar (20),
5     DataAssegnazione date,
6     DataFine date,
7     Tipo varchar(60),
8     Fisioterapista char (16) references Fisioterapisti(CF),
9     Ospite char (16) references Ospiti(CF),
10    primary key (Codice, DataAssegnazione,Ospite)
11 );
12
13 create table AssegnazioneAusiliAttuali
14 (
15     Codice char (4),
16     Contratto varchar (20),
17     DataAssegnazione date,
18     Tipo varchar(60),
19     Fisioterapista char (16) references Fisioterapisti(CF),
20     Ospite char (16) references OspitiInVita(CF),
21     primary key (Codice, DataAssegnazione,Ospite)
22 );
23
24
25 insert into StoricoAssegnazioneAusili values
26 ('0001','di propiet','14-JAN-2022','13-JAN-2023','carrozzina','GPACGL62D08E336B','DXLSPM52A02G600G'),
27 ('0002','di propiet','14-JAN-2021','13-JAN-2023','deaumbolatore','GPACGL62D08E336B','DXLSPM52A02G600G'),
28 ('0001','di propiet','7-DEC-2022','12-JAN-2023','carrozzina','GPACGL62D08E336B','DXLSPM52A02G600G');
29
30 insert into AssegnazioneAusiliAttuali values
31 ('0001','di propiet','14-JAN-2023','carrozzina','GPACGL62D08E336B','DXLSPM52A02G600G'),
32 ('0002','comodato','14-JAN-2023','deaumbolatore','GPACGL62D08E336B','DXLSPM52A02G600G'),
33 ('4567','di propiet','14-JAN-2023','carrozzina','GPACGL62D08E336B','ZBTFQF51L65F174I');
34
35

```

```

36 create table Pagamenti
37 (Data date,
38 Ospite char (16) references Ospiti (CF),
39 NumeroFattura char (4),
40 Importo int,
41 DipendenteAmministrativo char (16) references DipendentiAmministrativi(CF)
42 );
43
44
45 insert into Pagamenti values
46 ('14-JAN-2023', 'DXLSPM52A02G600G', '4567', 1987, 'NBRGKJ37D18A704J'),
47 ('14-DEC-2022', 'DXLSPM52A02G600G', '4533', 1987, 'NBRGKJ37D18A704J'),
48 ('14-NOV-2022', 'DXLSPM52A02G600G', '4566', 1987, 'NBRGKJ37D18A704J'),
49 ('14-OCT-2022', 'DXLSPM52A02G600G', '4562', 1987, 'NBRGKJ37D18A704J'),
50 ('14-JAN-2023', 'ZBTFQF51L65F174I', '4567', 1456, 'YVGZMD57D42I164J'),
51 ('14-DEC-2022', 'ZBTFQF51L65F174I', '4533', 1234, 'NBRGKJ37D18A704J'),
52 ('14-NOV-2022', 'ZBTFQF51L65F174I', '4566', 1456, 'YVGZMD57D42I164J'),
53 ('14-OCT-2022', 'ZBTFQF51L65F174I', '4562', 1345, 'NBRGKJ37D18A704J'),
54 ('14-JAN-2021', 'RVSNVK90B07I122X', '2566', 675, 'NBRGKJ37D18A704J'),
55 ('14-DEC-2021', 'RVSNVK90B07I122X', '2533', 876, 'NBRGKJ37D18A704J'),
56 ('14-NOV-2021', 'RVSNVK90B07I122X', '2566', 987, 'NBRGKJ37D18A704J'),
57 ('14-OCT-2020', 'JCVHSS97E48B744F', '2562', 1987, 'NBRGKJ37D18A704J'),
58 ('14-JAN-2020', 'JCVHSS97E48B744F', '2567', 1456, 'YVGZMD57D42I164J'),
59 ('14-DEC-2020', 'JCVHSS97E48B744F', '2533', 1234, 'NBRGKJ37D18A704J');

```

## 6.2 Definizioni di query

### 6.2.1 Query 1

Si decide di organizzare una festa di compleanno ogni ultimo giorno del mese, dove vengono festeggiati tutti coloro che hanno compiuto gli anni nel mese precedente. Viene quindi richiesto di restituire la lista dei compleanni di febbraio.

```
1 select nome,cognome
2 from ospiti
3 where extract (month from datanascita) = 2
```

nome	cognome
"Arsenio"	"Lo Duca"

### 6.2.2 Query 2

Query 2.1A fini statistici restituire il numero di ospiti accolti e deceduti durante lo scorso anno (2022).

```
1 select count(ospiti.CF) as OspitiAccoltiNel2022, count(ospitideceduti.CF) as OspitiDecedutiNel2022
2 from ospiti natural join ospitideceduti
3 where EXTRACT (YEAR from CURRENT_date)-extract (year from dataaccoglienza)=1
4 or EXTRACT (YEAR from CURRENT_date)-extract (year from datadecesso)=1
```

OspitiAccoltiNel2022	OspitiDecedutiNel2022
2	2

Query 2.2A fini statistici restituire il numero di ospiti accolti e deceduti nell'ultimo anno.

```
1 select count(*) as OspitiAccoltiNel2022
2 from ospiti
3 where CURRENT_date dataaccoglienza < 365
4 select count(*) as OspitiDecedutiNel2022
5 from ospitideceduti
6 where CURRENT_date datadecesso < 365
```



### 6.2.3 Query 3

Si vuole risalire agli ospiti sanissimi, quelli in vita mai ricoverati in ospedale e che non hanno mai avuto prescrizioni di farmaci.

```
1
2 --Per i sanissimi trovare gli ospiti per i quali
3 --non esiste neanche una diagnosi con collegata
4 --una prescrizione.
5
6 select cf, nome,cognome
7 from ospiti
8 where cf not in (select cfospite
9                  from ricoveriattuali)
10 and cf not in (select cfospite
11               from ricoveripassati)
12 and cf not in (select ospite
13               from diagnosi natural join prescrizioneifarmaci)
```

cf	nome	cognome
"ZBTFQF51L65F174I"	"Dorotea"	"Lombardi"
"JCVHSS97E48B744F"	"Ettore"	"Cociani"

Si noti che l'ospite 'ZBTFQF51L65F174I' ha effettuato una diagnosi ma questa non ha riportato la necessità di prescrizioni, di conseguenza non sarà tra i valori restituiti dalla terza query annidata.

### 6.2.4 Query 4

Si chiede di restituire gli ospiti che non sono in stanza con un ospite con il loro stesso nome.

```
1 select distinct ospite1.cf,ospite1.nome,ospite1.cognome
2 from ospiti as ospite1 cross join ospiti as ospite2
3 where ospite1.numerostanza=ospite2.numerostanza
4 and ospite1.cf!=ospite2.cf
5 and ospite1.nome=ospite2.nome
```

cf	nome	cognome
"JCVHSS97E48B744F"	"Ettore"	"Cociani"
"RVSNVK90B07I122X"	"Ettore"	"Siciliano"

### 6.2.5 Query 5

Si chiede di restituire gli ospiti attualmente positivi al COVID.

```
1 --Si cercano gli ospiti il cui ultimo tampone ha
2 --avuto esito positivo.
3 select cf,nome,cognome
4 from ospiti join tamponieffettuatiospiti as t1 on CF=Ospite
5 where data=(select max(data)
6              from tamponieffettuatiospiti
7              where t1.ospite=tamponieffettuatiospiti.ospite)
8 and esito
```

cf	nome	cognome
"DXLSPM52A02G600G"	"Arsenio"	"Lo Duca"

### 6.2.6 Query 6

Un ospite è positivo al COVID, per poter avvisare tutti coloro che sono stati in struttura a fare visita a un parente, è richiesto il nome e il numero di telefono dei parenti che sono stati in visita delle ultime due settimane.

```

1 select distinct nome,cognome,numeroditelefono
2 from parenti join visite on CF=CFParente
3 where current_date - data <15

```

cf	nome	cognome
"Artemisia"	"Trentini"	"0317 7605808"
"Lidia"	"Gallo"	"0324 5871722"

### 6.2.7 Query Medico

Controllare di aver effettuato almeno una visita a tutti gli ospiti in carico nell'ultimo mese.

```

1 select ospiti.nome as nomeospite, ospiti.cognome as cognomeospite, ospite as cfospite, medico as cfmedico
2 from rapportimedici join medici on medico=medici.cf
3 join ospiti on ospite=ospiti.cf
4 where (medico,ospite) not in (select medico,ospite
5                               from visitemediche
6                               where current_date- data < 30)

```

nomeospite	cognomeospite	cfospite	cfmedico
"Dorotea"	"Lombardi"	"ZBTFQF51L65F174I"	"SFCRFR52L06E952I"

### 6.2.8 Query Infermiere 1

Restituire per gli ospiti in vita i farmaci prescritti, in modo da sapere cosa va somministrato.

```

1 --Cerchiamo tutti i farmaci attualmente prescritti
2 --con la cadenza prevista e lultima volta in cui
3 --sono stati somministrati, in modo che si possa
4 --verificare se il momento di unaltra somministrazione.
5 select nome,cognome,codicefarmaco, ora AS oraultimasomministrazione, data AS dataultimasomministrazione, cadenza
6 from ospiti join diagnosi on cf=ospite
7 natural join prescrizioneifarmaci natural join somministrazionefarmaci as s1
8 where prescrizioneifarmaci.attiva
9       and data= (select max(data)
10                  from somministrazionefarmaci
11                  where s1.codicefarmaco=somministrazionefarmaci.codicefarmaco)

```

nome	cognome	codicefarmaco	oraultimasomministrazione	dataultimasomministrazione	cadenza
"Ettore"	"Siciliano"	"3456 "	"18:10:22"	"2023-01-15"	"6 ore"
"Ettore"	"Siciliano"	"5678 "	"04:10:25"	"2023-01-15"	"3 ore"
"Arsenio"	"Lo Duca"	"3256 "	"07:10:22"	"2023-01-14"	"10 ore"

### 6.2.9 Query Infermiere 2

Restituire ospiti che non hanno effettuato tamponi COVID nell'ultimo mese.

```
1 select Ospiti.cf, ospiti.nome, ospiti.cognome
2 from TamponiEffettuatiOspiti join ospiti on ospite=cf
3 where current_date-data >30
```

cf	nome	cognome
"DXLSPM52A02G600G"	"Arsenio"	"Lo Duca"
"ZBTFQF51L65F174I"	"Dorotea"	"Lombardi"

### 6.2.10 Query OSS

Per ogni ospite in struttura, restituire le prestazioni da fare: le mansioni giornaliere non ancora svolte nelle ultime 24 ore, le mansioni settimanali non ancora svolte negli ultimi 7 giorni.

```
1 select cf,nome,cognome,codicemansione
2 from ComposizionePiani join PianiAssistenza on CodicePiano=Codice
3 join Ospiti on ospite=cf join mansioni on Codicemansione=mansioni.codice
4 where mese='Gennaio'
5     and (mansioni.cadenza='giornaliera'
6         and (mansioni.codice,ospite) not in (select codicemansione,ospite
7                                             from prestazioni join mansioni on codicemansione=codice
8                                             and cadenza='giornaliera'
9                                             where data=current_date
10                                              or (current_date-data=1
11                                                  and extract (hour from current_time)
12                                                  - extract (hour from ora)<0)))
13     or (mansioni.cadenza='settimanale'
14         and (mansioni.codice,ospite) not in (select codicemansione,ospite
15                                             from prestazioni
16                                             join mansioni on codicemansione=codice
17                                             and cadenza='settimanale'
18                                             where current_date-data < 31))
```

Se si decidesse semplicemente di lavorare sulla tabella Prestazioni estraendo quelle giornaliere mai fatte nell'ultimo giorno e settimanali mai fatte nell'ultimo mese, si perdono le informazioni relative alle mansioni che non sono mai diventate prestazioni (in altre parole, se nel piano assistenziale è presente una mansione che non è mai stata attuata, questa non sarai mai presente nelle prestazioni e di conseguenza non viene restituita da una query che lavora solo sulla tabella Prestazioni). Si deve perciò risalire a tutte le mansioni prescritte nei piani e verificare che queste siano diventate prestazioni e che siano state fatte con la cadenza prevista.

cf	nome	cognome	codicemansione
"DXLSPM52A02G600G"	"Arsenio"	"Lo Duca"	"0001"
"DXLSPM52A02G600G"	"Arsenio"	"Lo Duca"	"0002"
"ZBTFQF51L65F174I"	"Dorotea"	"Lombardi"	"0002"

### 6.2.11 Query Fisioterapista

Per ogni ospite in struttura ottenere il numero di ausili prescritti attualmente e in passato.

```
1 select nome,cognome,codice
2 from ospiti join AssegnazioneAusiliAttuali on cf=ospite
```

nome	cognome	codice
"Arsenio"	"Lo Duca"	"0001"
"Arsenio"	"Lo Duca"	"0602"
"Dorotea"	"Lombardi"	"4567"

### 6.2.12 Query Operatore Amministrativo

Per ogni ospite in struttura, quanto ha pagato nel corso del suo soggiorno in struttura.

```
1 select sum(importo) as totalepagato,ospite
2 from pagamenti
3 group by ospite
```

totalepagato	ospite
7948	"DXLSPM52A02G600G"
4677	"JCVHSS97E48B744F"
2538	"RVSNVK90B07I122X"