

# Progetto Basi di Dati

Eros Fabrici M: 114087, Lorenzo Pittia M:

September 1, 2019

## Indice

<b>1</b>	<b>Analisi dei requisiti</b>	<b>2</b>
1.1	Testo esercizio . . . . .	2
1.2	Analisi requisiti . . . . .	2
<b>2</b>	<b>Modello E/R, vincoli di integrità e regole di derivazione</b>	<b>3</b>
2.1	Vincoli d'integrità . . . . .	3
2.2	Regole di derivazione . . . . .	3
2.3	Note sul modello ER . . . . .	4
<b>3</b>	<b>Progettazione Logica</b>	<b>5</b>
3.1	Analisi delle Ridondanze . . . . .	5
3.1.1	Tabella dei Volumi . . . . .	5
3.1.2	Tabella delle operazioni . . . . .	5
3.1.3	Analisi . . . . .	6
3.2	Eliminazione delle generalizzazioni . . . . .	8
3.3	Eliminazione degli attributi multi-valore . . . . .	8
3.4	Modello ER ristrutturato . . . . .	8
3.5	Modello relazionale . . . . .	9
3.6	Note sul modello relazionale e i trigger . . . . .	9
3.6.1	Normalizzazione . . . . .	10
<b>4</b>	<b>Indici</b>	<b>10</b>
<b>5</b>	<b>Analisi dei dati</b>	<b>10</b>
<b>6</b>	<b>Note finali</b>	<b>12</b>

# 1 Analisi dei requisiti

## 1.1 Testo esercizio

Si voglia modellare il seguente insieme di informazioni riguardanti i pazienti ricoverati in un dato ospedale italiano.

- Ogni reparto sia identificato univocamente dal nome, occupi uno e un solo piano di un determinato edificio e abbia un unico primario e uno o più altri medici. Si tenga traccia del numero di camere, del numero di letti disponibili e del numero di letti occupati di ogni reparto.
- Ogni medico, primari compresi, sia identificato univocamente dal codice fiscale e sia caratterizzato da un nome, un cognome, dall'anno di nascita e da una o più specializzazioni. Si assuma che ogni medico, primari compresi, possa lavorare per un solo reparto.
- Ogni camera di un dato reparto sia identificata da un numero (camera 5 del reparto Oculistica) e sia caratterizzata dal numero di letti disponibili e dal numero di letti occupati.
- Ogni letto di un dato reparto sia identificato da un numero (letto 21 del reparto Ginecologia) e sia caratterizzato dalla camera ove correntemente si trova. Si assuma che un letto possa essere nel tempo spostato da una camera ad un'altra di uno stesso reparto, ma non possa cambiare reparto.
- Ogni paziente sia identificato univocamente dal codice fiscale e sia caratterizzato dal nome, dal cognome, dall'anno di nascita e dal letto occupato. Si tenga presente che fra i pazienti vi possono essere anche medici dell'ospedale, primari compresi.

## 1.2 Analisi requisiti

Dall'analisi del testo è possibile capire che lo scopo principale del database è quello di conservare le informazioni dei pazienti di un ospedale italiano.

Dunque il database risulta essere indirizzato all'uso da parte del personale interno dell'ospedale, che si occupa di inserire i nuovi pazienti, aggiornare tutte le informazioni relative ad essi (che letto occupa, quanti letti disponibili, etc.), aggiungere le diagnosi effettuate dai medici, aggiungere i medici appena assunti, stampare le informazioni relative a i pazienti, visualizzare il numero di letti disponibili nei reparti, etc.

E' stato ritenuto necessario aggiungere (dato lo scopo del database), oltre a tutte le informazioni indicate nel testo, la possibilità di tenere traccia dei ricoveri passati di ogni paziente, quindi quando un paziente viene dimesso, esso non viene cancellato. Inoltre si è voluto tenere traccia delle diagnosi effettuate dai medici relative ai pazienti (ovviamente, siccome un medico può essere anche un paziente, non è possibile eseguire l'auto-diagnosi).

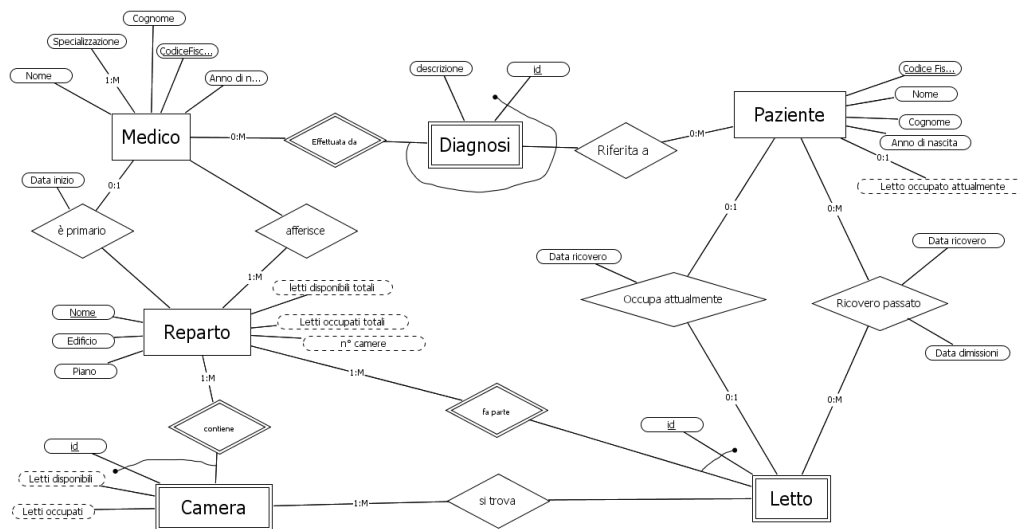
Dunque avremo le seguenti entità: *Medico*, *Paziente*, *Reparto*, *Camera* e *Letto*. Inoltre, dal testo è possibile osservare che l'entità *Paziente* e l'entità *Medico* hanno gli stessi attributi ad eccezione per il secondo che ne ha uno in più. Ciò può portare a fare uso di una generalizzazione, ma non è stato ritenuto utile, perché le due entità in questione non hanno relazioni in comune con le altre e quindi non sarebbe di alcuna utilità.

E' stato deciso di modellare, per semplicità, l'edificio e il piano come attributi di reparto, e non come un'entità separata, dato che in un edificio ci possono essere più reparti, così come in un piano.

## 2 Modello E/R, vincoli di integrità e regole di derivazione

Abbiamo usato un approccio inside-out, dato che è quello su cui abbiamo fatto più pratica a lezione. Siamo partiti dall'entità Medico, per poi espanderci ci siamo permessi di aggiungere un'entità, le relazione che ne conseguivano e una relazione in più per tenere traccia dei ricoveri passati. Vi sono due assunzioni che abbiamo preso in considerazione, come accennato precedentemente nell'analisi:

1. Un paziente può non avere una diagnosi (solo pazienti appena ricoverati, pazienti passati hanno sicuramente almeno una diagnosi).
2. Un primario, partecipa sia alla relazione è primario, che alla relazione afferisce.
3. Un medico non può diagnosticare un paziente di un reparto a cui non afferisce.
4. Un medico non può effettuarsi l'auto-diagnosi.



### 2.1 Vincoli d'integrità

1. Un Letto partecipa alla relazione "si trova" con una Camera la quale sta nella stesso Reparto del Letto (ergo: un letto facente parte di un certo Reparto può trovarsi soltanto in una Camera che fa parte del reparto in questione).
2. Un Paziente può essere anche Medico (e anche primario) e viceversa.
3. Un Paziente deve partecipare ad almeno una delle due seguenti relazioni: "occupa attualmente" o "ricovero passato".
4. Un medico non può effettuarsi l'autodiagnosi.
5. Un medico non può effettuare la diagnosi ad un paziente facente parte di un altro reparto rispetto a quello di appartenenza del primo.

### 2.2 Regole di derivazione

1. L'attributo derivato "n°camere" dell'entità Reparto viene derivato dalle partecipazioni con la relazione "contiene".

2. Gli attributi "letti disponibili" e "letti occupati" di *Camera* sono derivati dalla relazione "si trova".
3. Gli attributi "letti disponibili" e "letti occupati" dell'attributo *Reparto* viene derivato dalle partecipazioni con la relazione "contiene".
4. L'attributo derivato "letto occupato attualmente" di *Paziente* viene derivato dalla partecipazione con la relazione "occupa attualmente".

## 2.3 Note sul modello ER

Per quanto riguarda le entità *Camera* e *Letto* è stata presa in considerazione la possibilità di impostare la seconda come entità debole della prima e quest'ultima come entità debole di *Reparto*, ma non avrebbe rispettato il testo dell'esercizio. La soluzione presentata quindi crea un ciclo, il quale però non presenta problematiche che richiedono l'utilizzo di trigger, bensì le chiavi esterne sono sufficienti.

Vi è presente un ciclo anche tra le entità *Medico* e *Reparto*, il quale renderà necessaria l'implementazione di un trigger per il controllo dell'integrità (non è permesso assegnare come primario del reparto X un medico afferente al reparto Y).

Il ciclo tra *Paziente* e *Letto* non presenta alcuna problematica.

### 3 Progettazione Logica

In questa sezione si procede con vari raffinamenti, presentati durante il corso, al modello ER di modo da poterne trarre un modello logico equivalente.

#### 3.1 Analisi delle Ridondanze

##### 3.1.1 Tabella dei Volumi

La seguente tabella dei volumi è ipotetica e i dati che poi saranno effettivamente inseriti per testare il database non la rispecchieranno pienamente. Le decisioni di design ed implementative successive sono state dunque prese in base ai dati indicati nelle tabelle sottostanti.

Concetto	Tipo	Volume
Medico	E	500
Reparto	E	15
Camera	E	250
Letto	E	1000
Paziente	E	20000
Diagnosi	E	19600
E' primario	R	15
Afferisce	R	500
Contiene	R	250
Effettuata da	R	19500
Fa parte	R	1000
Si trova	R	1000
Occupi attualmente	R	700
Ricovero passato	R	19600
Riferita a	R	19600

Table 1: Tabella dei volumi

##### 3.1.2 Tabella delle operazioni

Operazioni	Frequenza
Inserimento nuovo paziente	100 al giorno
Inserimento nuovo medico	2 al mese
Inserimento nuova diagnosi	20 al giorno
Modifica stato paziente (da occupa attualmente a ricovero passato)	10 al giorno
Stampa informazioni di un paziente	100 al giorno
Stampa letti disponibili di un reparto	100 al giorno
Stampa letti disponibili di una camera	50 al giorno

Table 2: Tabella delle operazioni

### 3.1.3 Analisi

A seguire alcune delle operazioni che coinvolgono tutti gli attributi ridondanti:

**Operazione 1:** stampa letti disponibili di un reparto.

**Operazione 2:** stampa informazioni paziente (compreso id del letto occupato).

**Operazione 3:** stampa informazioni camera (compreso n°letti occupati e disponibili).

**Operazione 4:** inserimento di un nuovo paziente.

**Operazione 5:** stampa il numero di di camere di un dato reparto.

#### CON RIDONDANZA

Concetto	Tipo	N°accessi	Tipo accesso
Reparto	E	1	R

Table 3: Tabella accessi Operazione 1

Concetto	Tipo	N°accessi	Tipo accesso
Paziente	E	1	R

Table 4: Tabella accessi Operazione 2

Concetto	Tipo	N°accessi	Tipo accesso
Camera	E	1	R

Table 5: Tabella accessi Operazione 3

Concetto	Tipo	N°accessi	Tipo accesso
Paziente	E	1	W
Occupa attualmente	R	1	R
Occupa attualmente	R	1	W
Paziente	E	1	W
Letto	E	1	R
Fa parte	R	1	R
Reparto	E	1	W
Si trova	R	1	R
Camera	E	1	W

Table 6: Tabella accessi Operazione 4

Concetto	Tipo	N°accessi	Tipo accesso
Reparto	E	1	R

Table 7: Tabella accessi Operazione 5

## SENZA RIDONDANZA

Concetto	Tipo	N°accessi	Tipo accesso
Reparto	E	1	R
Fa parte	R	66	R
Letto	E	66	R
Occupa attualmente	R	66	R

Table 8: Tabella accessi Operazione 1

Concetto	Tipo	N°accessi	Tipo accesso
Paziente	E	1	R
Occupa attualmente	R	1	R
Letto	E	1	R

Table 9: Tabella accessi Operazione 2

Concetto	Tipo	N°accessi	Tipo accesso
Camera	E	1	R
Si trova	R	4	R
Letto	E	4	R
Occupa attualmente	R	4	R

Table 10: Tabella accessi Operazione 3

Concetto	Tipo	N°accessi	Tipo accesso
Paziente	E	1	W
Occupa attualmente	R	1	R
Occupa attualmente	R	1	W

Table 11: Tabella accessi Operazione 4

Concetto	Tipo	N°accessi	Tipo accesso
Reparto	E	1	R
Contiene	R	17	R

Table 12: Tabella accessi Operazione 5

E' possibile quindi osservare che la ridondanza riferita all'**operazione 1** sia necessaria per ridurre gli accessi. Per quanto riguarda l'**operazione 3**, in assenza di ridondanza non ha un costo considerevolmente maggiore rispetto al caso in cui vi sono presenti gli attributi in questione (una camera non avrà mai un numero potenzialmente elevato di letti, al massimo una decina), dunque è stata decisa la loro rimozione (attributi "letti disponibili" e "letti occupati" dell'entità *Camera*).

Relativamente l'attributo "letto occupato attualmente" di Paziente, riteniamo che non sia opportuno tenerlo, dato che un paziente occuperà al più 1 solo letto. L'**operazione 4** risulta leggermente più

efficiente se assenti gli attributi ridondanti, ma questo guadagno non è comparabile al costo che si avrebbe, come detto in precedenza, nell'esecuzione dell'**operazione 1**.

In base ai risultati dell'**operazione 5** è stata decisa la rimozione dell'attributo "n°camere" dato che da requisiti, il numero camere di un reparto rimane fisso.

L'esecuzione dell'**operazione 2** non presenta particolari benefici in termini di efficienza in presenza dell'attributo ridondante "letto occupato attualmente". Si è deciso dunque di rimuoverlo.

Ricapitolando:

1. Rimuovere l'attributo "letto occupato attualmente" dall'entità *Paziente*.
2. Conservare gli attributi "letti disponibili totali" e "letti occupati totali" dell'entità *Reparto* e rimuoverne l'attributo "n°camere".
3. Rimuovere gli attributi "letti disponibili" e "letti occupati" dell'entità *Camera*
4. Sono invece stati mantenuti gli attributi "letti occupati totali" e "letti disponibili totali" dell'entità *Reparto*.

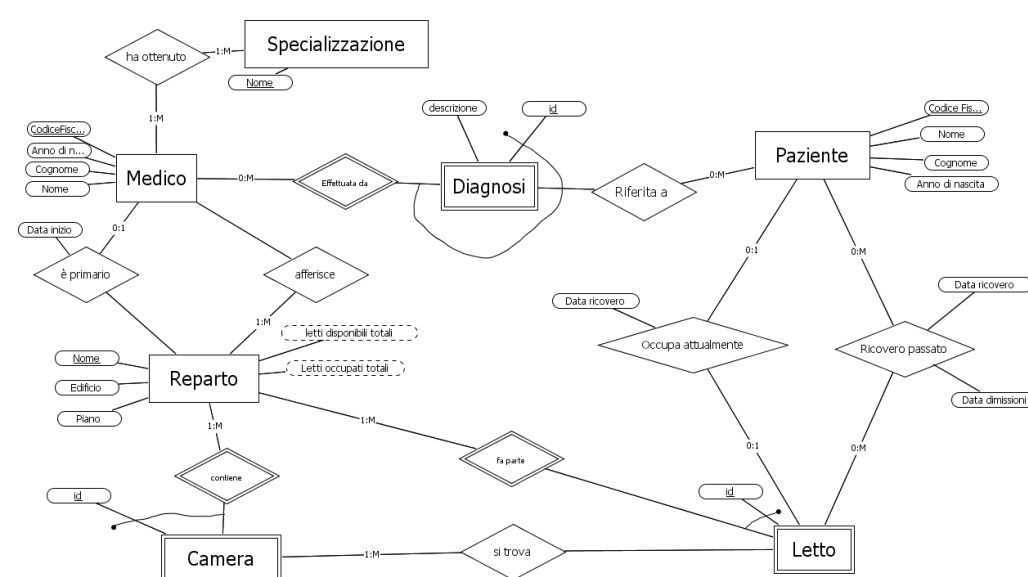
### 3.2 Eliminazione delle generalizzazioni

Non vi è presente nessuna generalizzazione, dunque si è passati direttamente al punto successivo.

### 3.3 Eliminazione degli attributi multi-valore

L'entità *Medico* ha l'attributo "specializzazione" che ha cardinalità 1:n. E' dunque necessario eliminare quell'attributo per sostituirlo con un'entità omonima e una relazione ("ha conseguito") che la lega con *Medico*. La nuova entità avrà gli attributi "Nome", che la identifica univocamente, e "Descrizione".

### 3.4 Modello ER ristrutturato





### 3.5 Modello relazionale

*medico*(cod\_fiscale, nome, cognome, anno\_di\_nascita)  
*reparto*(nome, edificio, piano, letti\_disponibili, letti\_occupati, primario, data\_inizio)  
*camera*(id, reparto)  
*letto*(id, reparto)  
*paziente*(cod\_fiscale, nome, cognome, anno\_di\_nascita)  
*specializzazione*(nome)  
*ha\_ottenuto*(medico, specializzazione)  
*afferisce*(medico, reparto)  
*si\_trova*(id\_letto, reparto, id\_camera)  
*occupa\_attualmente*(paziente, id\_letto, reparto, data\_ricovero)  
*ricovero\_passato*(paziente, data\_ricovero, data\_dimissioni, letto, reparto)  
*diagnosi*(id, medico, paziente, descrizione)

#### Chiavi esterne:

- *reparto*.primario → *medico*.cod\_fiscale
- *camera*.reparto → *reparto*.nome
- *letto*.reparto → *reparto*.nome
- *ha\_ottenuto*.medico → *medico*.cod\_fiscale
- *ha\_ottenuto*.specializzazione → *specializzazione*.nome
- *afferisce*.medico → *medico*.cod\_fiscale
- *afferisce*.reparto → *reparto*.nome
- (*si\_trova*.id\_letto, *si\_trova*.id\_reparto) → (*letto*.id, *letto*.reparto)
- (*si\_trova*.id\_camera, *si\_trova*.reparto) → (*camera*.id, *camera*.reparto)
- *occupa\_attualmente*.paziente → *paziente*.cod\_fiscale
- (*occupa\_attualmente*.id\_letto, *occupa\_attualmente*.reparto) → (*letto*.id, *letto*.reparto)
- *ricovero\_passato*.paziente → *paziente*.cod\_fiscale
- (*ricovero\_passato*.id\_letto, *ricovero\_passato*.reparto) → (*letto*.id, *letto*.reparto)
- *diagnosi*.medico → *medico*.cod\_fiscale
- *diagnosi*.paziente → *paziente*.cod\_fiscale

### 3.6 Note sul modello relazionale e i trigger

- La relazione è *primario* e il suo attributo presenti nel modello ER, sono stati rappresentati come due attributo di reparto, **primario** e **data\_inizio**, con aggiunta di trigger appositi per soddisfare i vincoli posti dal testo dell'esercizio.
- La relazione *ricovero\_passato* ha come unico trigger il controllo che la **data\_ricovero** sia minore o uguale alla **data\_dimissioni**. Ciò permetterebbe di inserire due ricoveri di un paziente con data di ricovero uguale e data di dimissioni differente, che risulta abbastanza complicato da gestire con i trigger. Assumiamo quindi che non vengano commessi errori del genere nell'inserimento dei dati.
- Dato uno degli scopi principali del database è quello di tenere lo storico dei pazienti, si suppone che i record della relazione *paziente* non vengano mai eliminati: quando un paziente viene dimesso, si elimina la tupla corrispondente in *occupa\_attualmente* e se ne deve aggiungere una

nuova manualmente in *ricovero\_passato* con i dati del paziente in questione.

- Il vincolo che riguarda la diagnosi, in particolare quello che un medico non può effettuare una diagnosi a un paziente di un altro reparto, si applica per i pazienti attualmente ricoverati: si suppone quindi che gli inserimenti di diagnosi di pazienti passati siano corretti.
- Il vincolo che impone l'impossibilità di un letto di essere spostato in una camera di un reparto differente al quello di appartenenza viene rispettato grazie all'impostazione delle chiavi secondarie
  - $(si\_trova.id\_letto, si\_trova.id\_reparto) \rightarrow (letto.id, letto.reparto)$
  - $(si\_trova.id\_camera, si\_trova.reparto) \rightarrow (camera.id, camera.reparto)$

Non vede quindi la necessità di trigger.

### 3.6.1 Normalizzazione

Il modello rispetta tutte e tre le forme di normalizzazione.

- **1° Forma:** non esistono attributi ripetuti o composti;
- **2° Forma:** tutti gli attributi dipendono dalla chiave primaria completa;
- **3° Forma:** gli attributi dipendono solo dalla chiave primaria;

## 4 Indici

Gli indici hanno lo scopo di ottimizzare l'interazione fra utente e database.

La scelta è stata presa sulla base della tabella dei volumi e delle operazioni precedentemente presentate, dalle quali si può osservare che *ricovero\_passato* e *diagnosi* hanno una grande quantità di record e vi si accede con un'elevata frequenza (l'operazione "stampa informazioni di un paziente" include tutte le informazioni relative a un paziente, non solo quelle presenti nella tabella omonima), i cui attributi **paziente** si ripetono più volte all'interno delle singole tabelle.

Dunque, gli indici ottimali individuati sono due:

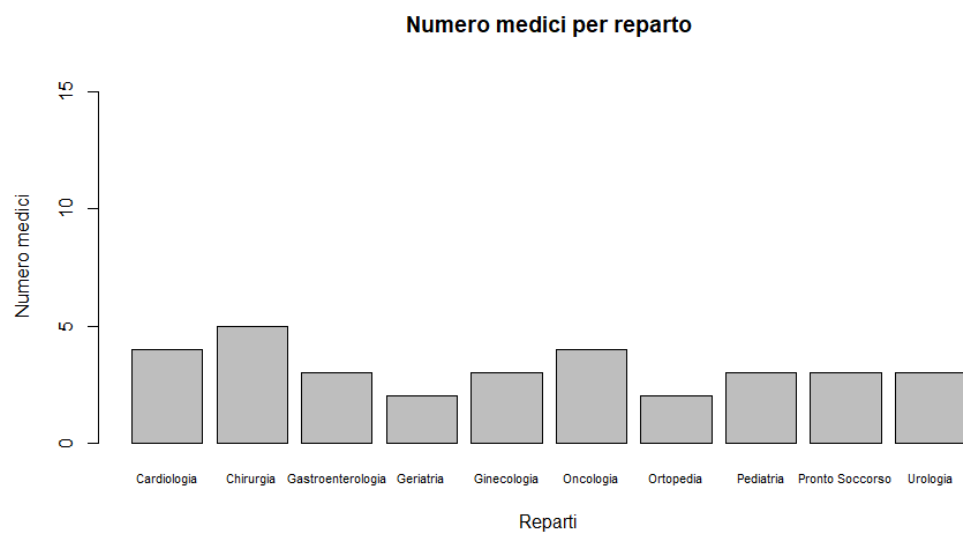
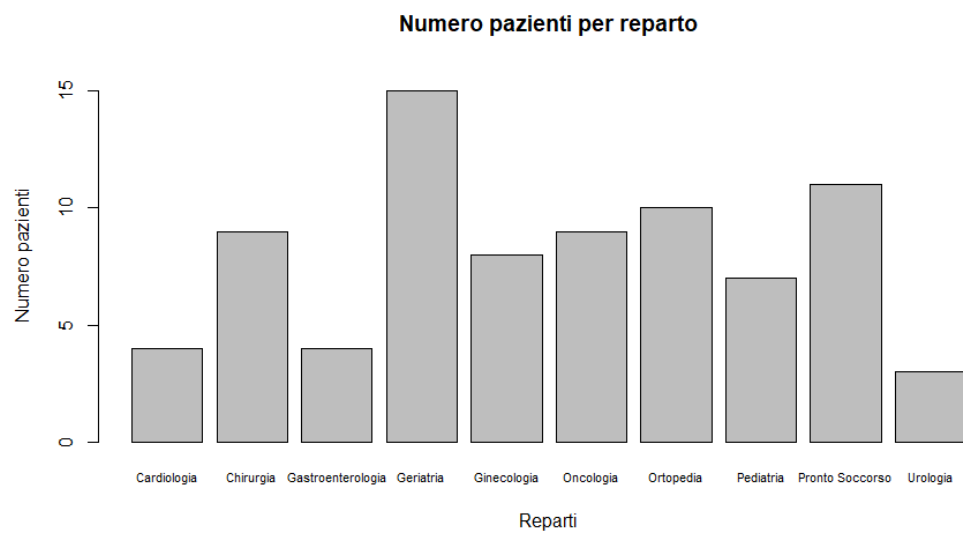
- Indice su *ricovero\_passato.paziente*
- Indice su *diagnosi.paziente*

In questo modo, possiamo ottimizzare la ricerca quando si voglia cercare tutti i ricoveri passati di un certo paziente o quando si vuole estrarre tutte le diagnosi effettuate su un determinato paziente.

## 5 Analisi dei dati

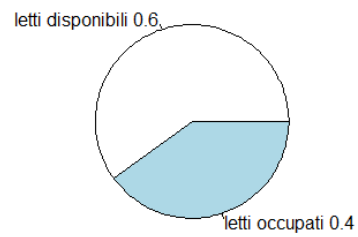
A seguire l'analisi dei dati eseguita con R.

Dai primi due barplots si può evidenziare che l'ospedale non ha sufficienti medici per far fronte alla quantità di pazienti attualmente ricoverati, evidenziando anche i reparti con più carenze (per esempio il reparto di Geriatria).



Il grafico a torta a seguire mostra il rapporto tra letti disponibili e letti occupati di tutto l'ospedale. Si può vedere come molti letti sono attualmente vuoti. L'ospedale potrebbe dunque gestire molti più pazienti, a patto di risolvere il problema della carenza di medici.

**Grafico a torta letti**



## 6 Note finali

- Il database è stato creato e testato con MariaDB (terminale).
- Tutto il codice sql (creazione, trigger, popolamento, indici, query esemplificative) si trova nel file db.sql.