



SAPIENZA  
UNIVERSITÀ DI ROMA

## Basi di Dati, Modulo 2

Sapienza Università di Roma

Facoltà di Ing. dell'Informazione, Informatica e Statistica

Laurea in Informatica

Prof. Toni Mancini

<http://tmancini.di.uniroma1.it>

Progetto 20080110 (P.20080110)

# QuickHospital

Versione 2024-02-08

# Indice

Indice	1
1 Introduzione	2
2 Specifica dei Requisiti	3

## 1

---

## Introduzione

Si vuole progettare e realizzare *QuickHospital*, un sistema informatico per la gestione di ricoveri e di visite mediche in un ospedale estremamente efficiente. Il sistema deve permettere la memorizzazione e gestione dei pazienti e dei relativi ricoveri ospedalieri e prenotazioni di visite ambulatoriali, nonché degli itinerari di visita dei medici dell'ospedale.

## 2

## Specifica dei Requisiti

Il sistema *QuickHospital* deve permettere di memorizzare e gestire informazioni circa i pazienti e i medici dell'ospedale nel quale viene installato. In particolare, dei pazienti interessano alcune informazioni anagrafiche (nome, cognome e data di nascita) ed i loro  
5 recapiti, distinti in recapiti telefonici, recapito email e postale (questi ultimi unici).

Per quanto riguarda i medici dell'ospedale invece, interessa mantenere informazioni sul loro nome, cognome e data di nascita, ed i pazienti che hanno in cura.

Un paziente può essere ricoverato, in una certa data, solo se una precedente verifica della disponibilità dei posti letto presenti nell'ospedale ha dato esito positivo.

10 Una volta effettuato il ricovero, il paziente ha assegnato un posto letto nell'ambito di una stanza; una stanza può contenere da un minimo di 1 ad un massimo di 8 posti letto. Le stanze hanno un piano ed un settore (interi positivi).

Il sistema deve inoltre permettere la memorizzazione dello storico di tutti i pazienti che sono stati ricoverati e poi dimessi nel tempo, con le informazioni relative ai posti  
15 letto occupati durante i diversi ricoveri.<sup>1</sup>

Sono funzionalità specifiche del sistema la registrazione del ricovero di un paziente e della sua dimissione ad opera del personale di accettazione. Inoltre il sistema deve assistere i medici ottimizzando il loro percorso di visite.

In particolare, il sistema deve permettere di calcolare, su richiesta di un medico, il  
20 suo itinerario delle visite, ovvero un insieme ordinato delle stanze cui accedere (che sono tutte e sole le stanze che ospitano i pazienti che ha in cura).

L'ordinamento è dato in primo luogo dal piano delle stanze dei pazienti da visitare, ed in secondo luogo dal settore di appartenenza di tali stanze (entrambi in ordine crescente). I settori sono infatti numerati secondo un criterio di vicinanza topologica. Pertanto se  
25 un dato medico deve visitare le stanze  $\{(7, 4), (7, 1), (1, 3), (1, 1), (3, 4)\}$  dove la prima componente di ognuna è il piano e la seconda il settore, l'itinerario di visita proposto deve essere  $[(1, 1), (1, 3), (3, 4), (7, 1), (7, 7)]$ .

<sup>1</sup>Si assuma per semplicità che durante il periodo di un ricovero il paziente non possa cambiare letto.

Oltre ai pazienti dell'ospedale, il sistema gestisce anche prestazioni mediche fatte da medici dell'ospedale a pazienti esterni. L'anagrafica di tali pazienti è registrata nel sistema (ad opera del personale addetto alle prenotazioni), con l'informazione aggiuntiva della particolare prestazione medica richiesta al personale ospedaliero (oltre che la data richiesta). Le prestazioni sono caratterizzate da una specializzazione richiesta (ad., ortopedia, dermatologia, ecc.) e una descrizione più estesa. 30

Di ogni medico il sistema deve conoscere la sua specializzazione primaria e le sue specializzazioni secondarie. 35

Data una prestazione richiesta da un paziente esterno (per una specializzazione  $s$ ), il sistema deve restituire l'insieme dei medici maggiormente idonei a soddisfarla. Il criterio di idoneità è il seguente: se esistono medici con specializzazione primaria pari ad  $s$ , il risultato è l'insieme di tali medici. Altrimenti, il risultato è l'insieme dei medici che hanno  $s$  tra le loro specializzazioni secondarie. 40

Il sistema *QuickHospital* è accessibile ai medici, al personale amministrativo e a quelli dell'ufficio prenotazioni.



**Esame BD2.Esame.Risposte – Modulo risposte prova scritta (diagramma delle classi UML)**

**Dati dello studente e dell'esame**

Cognome e nome: ..... Matricola: .....

Data: .....

Corso di laurea e canale di appartenenza:

- ☐ Laurea in Informatica, canale 1 (Prof. G. Perelli)  
☐ Laurea in Informatica, canale 2 (Prof.ssa M. De Marsico)

Firma di un membro della Commissione per  
avvenuta identificazione:

.....

**Rinuncia alla prova**

☐ Desidero rinunciare a questa prova d'esame. Firma: .....





# Istruzioni e regole d'esame

## Prima dell'esame

- Stampare questo modulo, preferibilmente fronte-retro, e rilegarlo con un fermaglio rimovibile, come quello disegnato in alto
- Compilare il frontespizio con i propri dati, come richiesto
- Scrivere la propria matricola nello spazio apposito nella parte alta di tutte le pagine

## Durante l'esame

- La prova è dimensionata per essere svolta in circa 3 ore. Tuttavia, data la sua natura fortemente progettuale, la Commissione offre agli studenti la più ampia disponibilità di tempo, al fine ovviare ad eventuali (e limitati) errori di analisi/progettazione rilevati più a valle del ciclo di vita.  
Il tempo massimo per la consegna è quindi rilassato a 5 ore (il massimo tempo compatibile con le disponibilità di aule).
- Scrivere le risposte negli spazi predisposti sotto le relative domande. Le ultime pagine sono vuote e possono essere usate come minute oppure, se puntate opportunamente, per contenere risposte in caso gli spazi appositi dovessero risultare insufficienti.
- Non è possibile usare alcun tipo di materiale didattico.
- In caso di necessità di ulteriori fogli (in proprio possesso), chiedere preventivamente alla Commissione una nuova procedura di controllo.
- La Commissione può rispondere solo a brevi domande inerenti al testo dei quesiti.
- Tra la seconda e la quarta ora d'esame, gli studenti possono effettuare **brevi pause** (uno studente alla volta) seguendo la seguente procedura:
  1. Alla lavagna è riportata una coda denominata 'Coda prenotazioni pause'. Sia  $n$  (un intero) l'elemento in fondo alla coda (si assuma  $n = 0$  in caso di coda vuota).
  2. Recarsi alla lavagna ed aggiungere l'intero  $n + 1$  come proprio contrassegno in fondo alla coda, seguito da una stringa a propria scelta (ad es., le proprie iniziali).
  3. Se il proprio contrassegno non è l'elemento affiorante della coda, tornare al lavoro in attesa che lo diventi.
  4. Consegnare tutti i fogli di lavoro e il testo d'esame alla Commissione ed uscire.
  5. Al rientro, cancellare il proprio contrassegno dalla coda di modo da permettere al successivo studente prenotato di uscire, e riprendere i fogli prima consegnati.

## Al momento della consegna

- Ordinare tutti i fogli che si vuole far valutare e rilegarli con un fermaglio rimovibile. Non includere fogli che la Commissione non deve valutare (ad es., requisiti, minute), ma includere ovviamente il frontespizio.
- Consegnare i fogli ordinati **nelle mani** di un membro della Commissione. **Non** lasciare l'aula senza la conferma, da parte della Commissione, del buon esito delle operazioni di consegna.

## In caso di rinuncia

- È possibile rinunciare alla consegna a partire dalla seconda ora d'esame. In caso di rinuncia, consegnare nelle mani della Commissione solo il frontespizio, dopo aver compilato e firmato la sezione dedicata.

## Sommario delle domande

Si richiede di progettare l'applicazione descritta dalla specifica dei requisiti effettuando le fasi di Analisi concettuale dei requisiti e di Progettazione logica della base dati e delle funzionalità, utilizzando la metodologia vista nel corso.

In particolare (vengono indicati i tempi suggeriti per i diversi passi chiave):

**Parte 1: Analisi concettuale dei requisiti** Effettuare la fase di Analisi concettuale dei requisiti producendo lo schema concettuale per l'applicazione, che includa:

- Analisi dei dati (45 minuti; 75 minuti al massimo):
  - un diagramma UML concettuale delle classi (\*)
  - (parte del)le specifiche formali delle classi e delle associazioni
  - le specifiche dei tipi di dato
  - la specifica formale dei vincoli esterni (\*)
- Analisi delle funzionalità:
  - un diagramma UML degli use-case (5 minuti; 10 minuti al massimo)
  - la segnatura di tutte le operazioni di use-case (10 minuti)
  - (parti del)le specifiche formali degli use-case. (30 minuti; 60 minuti al massimo)

Si richiede *esplicitamente* di modellare le specifiche formali delle operazioni di classe e/o use-case necessarie a modellare i requisiti contrassegnati dalla barra laterale (come quella qui a sinistra), *incluse* tutte le eventuali operazioni ausiliarie, usando l'estensione della logica del primo ordine studiata nel corso. (\*)

**Parte 2: Progettazione della base dati e delle funzionalità** Effettuare la progettazione della base dati e delle funzionalità a partire dallo schema concettuale prodotto nella Parte 1, ed in particolare eseguire i seguenti passi:

- Progettazione della base dati relazionale con vincoli:
  - Ristrutturazione del diagramma UML concettuale delle classi e delle specifiche (20 minuti; 30 minuti al massimo):
    - \* scelta del DBMS da utilizzare
    - \* progettazione della corrispondenza tra i tipi di dato concettuali ed opportuni domini SQL (domini base o utente, oppure realizzati mediante relazioni aggiuntive) supportati dal DBMS scelto
    - \* ristrutturazione del diagramma UML concettuale delle classi e delle specifiche dei vincoli esterni.
  - Produzione dello schema relazionale della base dati e dei relativi vincoli (\*) (30 minuti; 60 minuti al massimo)
- Progettazione delle funzionalità (30 minuti; 45 minuti al massimo):
  - definizione della specifica realizzativa delle operazioni necessarie a modellare i requisiti contrassegnati dalla barra laterale, in modo conforme alla loro specifica concettuale prodotta nella fase di Analisi, in termini di algoritmi in pseudo-codice e comandi SQL immersi. (\*)

Le pagine seguenti contengono le domande specifiche a cui è richiesto rispondere, ulteriori delucidazioni per ogni singolo punto, e spazi per le risposte.

Le pagine da 31 in poi possono essere utilizzate per scrivere minute che non verranno valutate.

---

(\*) Una risposta soddisfacente a questa domanda è condizione *necessaria* (ma non sufficiente) per superare la prova.





Questa pagina è stata intenzionalmente lasciata vuota

# 1 Analisi concettuale

**Domanda 1 (10 minuti)** Raffinare la specifica dei requisiti eliminando inconsistenze, omissioni e ridondanze e producendo un elenco numerato di requisiti il meno ambiguo possibile. (La risposta a questa domanda non sarà valutata, ma si consiglia di svolgere accuratamente questo passo, in quanto può facilitare di molto le attività di progetto.)

Risposta

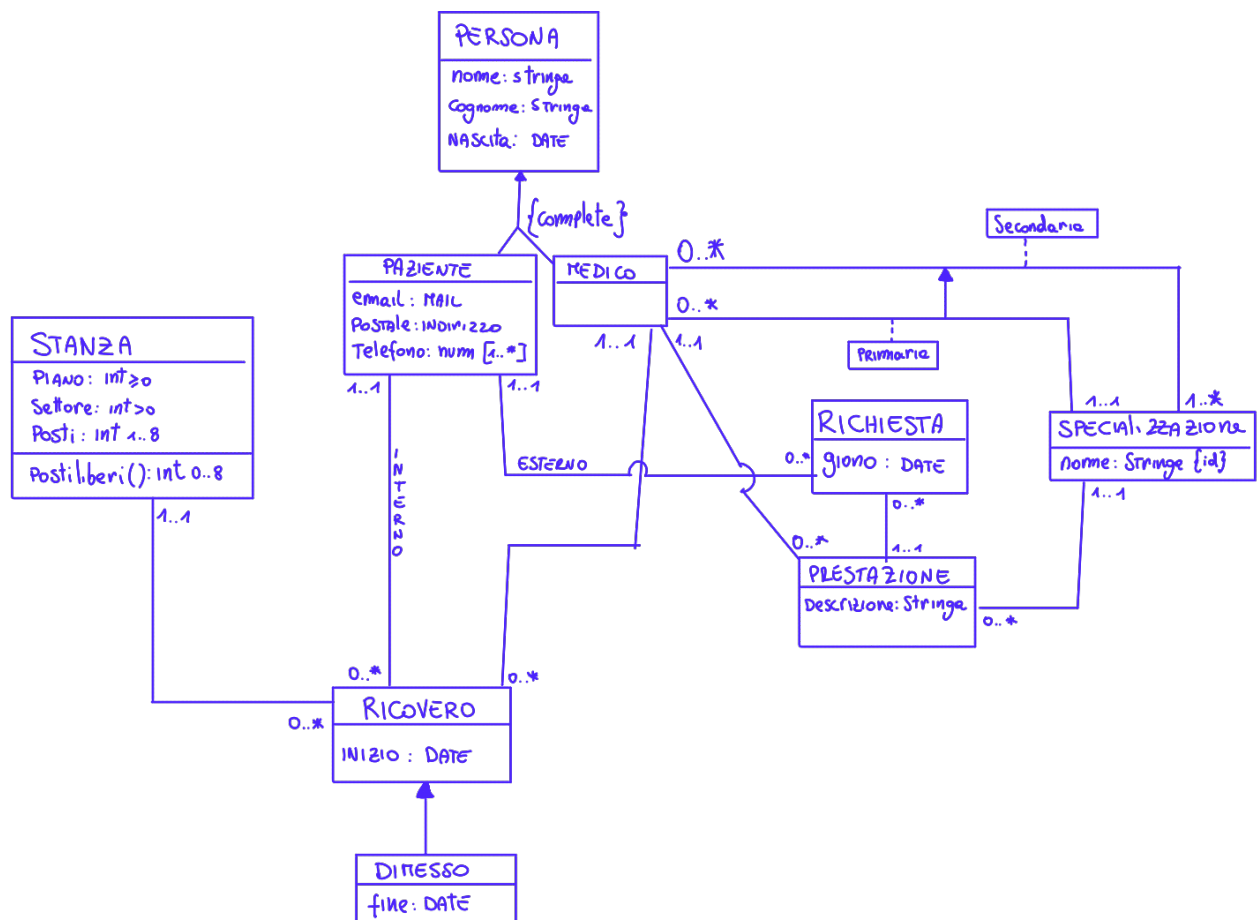
Risposta alla Domanda 1 (segue)

**Domanda 2 (45 minuti; 75 minuti al massimo)** Proseguire la fase di Analisi Concettuale dei requisiti, producendo un diagramma UML concettuale delle classi per l'applicazione, le specifiche di classi, associazioni, tipi di dato e vincoli esterni.

Una risposta soddisfacente a questa domanda è condizione *necessaria* (ma non sufficiente) per superare la prova.

### Diagramma UML concettuale delle classi

Produrre un diagramma UML concettuale delle classi per l'applicazione in termini di classi, associazioni, attributi, generalizzazioni, operazioni di classe.



Risposta alla Domanda 2 (segue)

**Specifiche delle classi o associazioni** Per ogni classe o associazione del diagramma **con** operazioni o vincoli:

- Definire la specifica formale di eventuali operazioni necessarie a modellare i requisiti contrassegnati dalla barra laterale, ed eventuali vincoli esterni. Usare la logica del primo ordine estesa con teoria degli insiemi e semantica di mondo reale vista nel corso, usando il seguente alfabeto:
  - Un simbolo di predicato  $C/1$  per ogni classà  $C$ .  
Semantica di  $C(x)$ :  $x$  è una istanza di  $C$ .
  - Un simbolo di predicato  $T/1$  per ogni tipo di dato  $T$ .  
Semantica di  $T(x)$ :  $x$  è un valore di  $T$ .
  - Un simbolo di predicato  $\text{assoc}/2$  per ogni associazione binaria  $\text{assoc}$ .  
Semantica di  $\text{assoc}(c_1, c_2)$ :  $(c_1, c_2)$  è una istanza di  $\text{assoc}$ .
  - Un simbolo di predicato  $\text{attr}/2$  per ogni attributo  $\text{attr}$  di entità  
Semantica di  $\text{attr}(c, v)$ : uno dei valori dell'attributo  $\text{attr}$  dell'istanza  $c$  è  $v$ .
  - Un simbolo di predicato  $\text{attr}/3$  per ogni attributo  $\text{attr}$  di associazione binaria.  
Semantica di  $\text{attr}(c_1, c_2, v)$ : uno dei valori dell'attr.  $\text{attr}$  del link  $(c_1, c_2)$  è  $v$ .
  - Un simbolo di predicato  $\text{op}/(n+2)$  per ogni operazione di classe ad  $n$  argomenti.  
Semantica di  $\text{op}(c, \arg_1, \dots, \arg_n, v)$ : uno dei valori di ritorno di  $\text{op}$ , quando invocata sull'istanza  $c$  e con argomenti  $\arg_1, \dots, \arg_n$  è  $v$ .
  - Il simbolo di  $=/2$  (la cui interpretazione è la relazione che lega ogni elemento del dominio di interpretazione solo con se stesso) e opportuni simboli di predicato e di funzione, soggetti a semantica di modo reale, per relazioni e funzioni standard tra elementi dei tipi di dato, tra cui  $\text{adesso}/0$ , interpretato come il valore del dominio DataOra che rappresenta l'istante corrente.

## Risposta

<p>1 Tipo: <b>Classe</b>   <b>Associazione</b> (cerchiare)</p> <p>Nome: .... <b>PAZIENTE</b> .....</p> <p>Operazioni, vincoli:</p> <p>Ⓢ [V. non si può ESSERE IL medico di SE STESSO]</p> <p><math>\forall x, r (\text{paz\_ric}(x, r) \rightarrow \neg \text{med\_ric}(x, r))</math></p>	<p>2 Tipo: <b>Classe</b>   <b>Associazione</b> (cerchiare)</p> <p>Nome: .... <b>DISMESSO</b> .....</p> <p>Operazioni, vincoli:</p> <p>Ⓢ <math>\forall d, i, f \text{ dismesso}(d) \wedge \text{inizio}(d, i) \wedge \text{fine}(d, f) \rightarrow i \leq f</math></p>
---	---

<p>3 Tipo: <b>Classe</b>   <b>Associazione</b> (cerchiare)</p> <p>Nome: ... <b>PRESTAZIONE</b> .....</p> <p>Operazioni, vincoli:</p> <p>Ⓣ [V. impedisce a medici non competenti di svolgere prestazioni]</p> $\forall p, s (\text{prestazione}(p) \wedge \text{specializzazione}(s) \wedge \text{pre\_spe}(p, s)) \longrightarrow \neg \exists m (\text{medico}(m) \wedge \text{med\_pre}(m, p) \wedge \neg \text{med\_spe}(m, s))$	<p>6 Tipo: <b>Classe</b>   <b>Associazione</b> (cerchiare)</p> <p>Nome: ... <b>STANZA</b> .....</p> <p>Operazioni, vincoli:</p> <p>Ⓣ [V. numero ricoverati minore dei posti]</p> $\forall s, p (\text{stanza}(s) \wedge \text{posti}(s, p) \longrightarrow  \{r   \text{sta\_ric}(s, r)\}  \leq p)$
<p>4 Tipo: <b>Classe</b>   <b>Associazione</b> (cerchiare)</p> <p>Nome: ... <b>STANZA</b> .....</p> <p>Operazioni, vincoli:</p> <p>PostiLiberi(): intero 0..8</p> <p>PRE:</p> <p>Post:</p> $L =  \{r   \text{ricovero}(r) \wedge \text{sta\_ric}(\text{this}, r)\} $ <p>Sia P := Posti(this, p)</p> <p>RESULT = P - L</p>	<p>7 Tipo: <b>Classe</b>   <b>Associazione</b> (cerchiare)</p> <p>Nome: ... <b>RICOVERO</b> .....</p> <p>Operazioni, vincoli:</p> <p>Ⓣ [V. no overlaps ricoveri]</p> $\forall p, r', i', f'$ $(\text{paziente}(p) \wedge \text{ricovero}(r') \wedge \text{inizio}(r', i') \wedge \text{fine}(r', f') \wedge \text{Paz\_ric}(p, r'))$ $\longrightarrow \neg \exists r'', i'', f'' (\text{ricovero}(r'') \wedge \text{inizio}(r'', i'') \wedge \text{fine}(r'', f'') \wedge \neg (f'' \leq i' \vee i'' \geq f')) \wedge (p, r'')$
<p>5 Tipo: <b>Classe</b>   <b>Associazione</b> (cerchiare)</p> <p>Nome: .....</p> <p>Operazioni, vincoli:</p>	<p>8 Tipo: <b>Classe</b>   <b>Associazione</b> (cerchiare)</p> <p>Nome: .....</p> <p>Operazioni, vincoli:</p>

Specifiche dei tipi di dato, specifiche di ulteriori vincoli esterni ed altre specifiche

tipi di dato

MAIL = ... ← Requisiti email

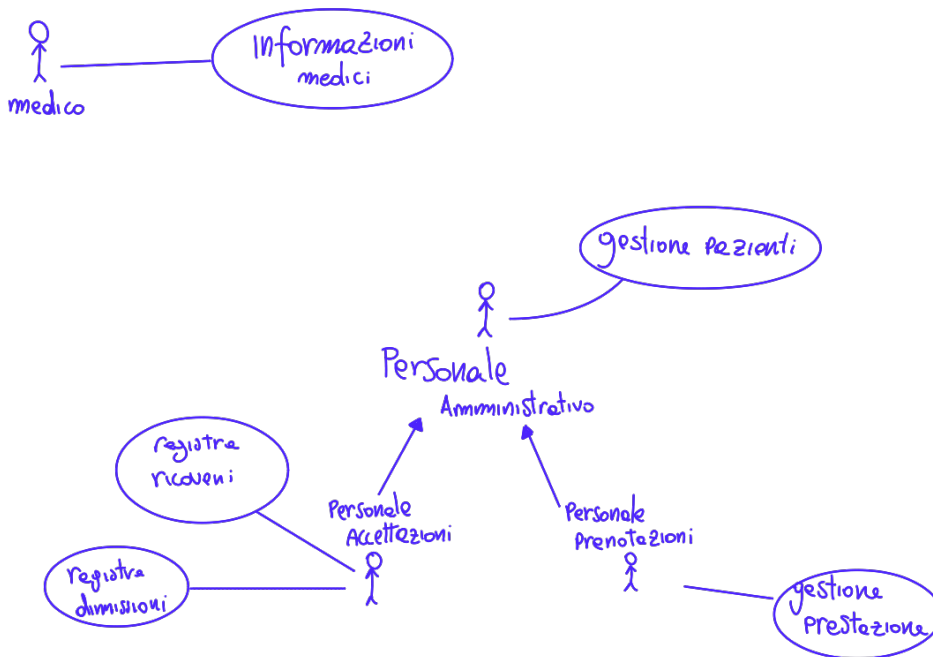
Indirizzo = (via: stringa, civico: int > 0)

num = [0...9]{10} ← Requisiti numero telefono



Risposta alla Domanda 2 (segue)

**Domanda 3 (5 minuti; 10 minuti al massimo)** Proseguire la fase di Analisi Concettuale dei requisiti, producendo un diagramma UML degli use-case che definisca ad alto livello tutte le funzionalità richieste al sistema.

**Risposta**



Questa pagina è stata intenzionalmente lasciata vuota

**Domanda 4 (10 minuti)** Proseguire la fase di Analisi Concettuale dei requisiti definendo la **segnatura** delle operazioni in ogni use-case.

**Risposta**

def. ITINERARIO VISITE

visite(m: medico): STANZA [0..\*]

Registra ricoveri

agg-ricovero(p: INTERNO, i: DATE): RICOVERO

Registra dimissioni

agg-dimissione(r: ricovero, f: DATE): DIMISSIONE

Registra PAZIENTI

agg-ESTERNI(nome: stringa, ...): PAZIENTE

def. Medici[donei (p: PRESTAZIONE): Medici [0..\*]

Registra PRESTAZIONE

agg-PRESTAZIONE(r: richiesta, desc: stringa): PRESTAZIONE



Questa pagina è stata intenzionalmente lasciata vuota

**Domanda 5 (30 minuti; 60 minuti al massimo)** Proseguire la fase di Analisi Concettuale dei requisiti producendo le specifiche concettuali per le operazioni di use-case, **limitandosi** a quelle necessarie a modellare i requisiti contrassegnati dalla barra laterale (come quella qui a sinistra), ed includendo eventuali operazioni ausiliarie. In particolare, per ogni operazione, definire segnatura, precondizioni e postcondizioni utilizzando il linguaggio della logica del primo ordine. Si assuma lo stesso vocabolario definito alla **Domanda 2**.

Una risposta soddisfacente a questa domanda è condizione *necessaria* (ma non sufficiente) per superare la prova.

### Risposta

$\text{mediciIDONEi} (p: \text{PRESTAZIONE}): \text{Medico} [0..*]$

• PRE:

• POST:

Sia  $S \mid \text{PRE.SPE}(p, s) \wedge \text{SPECIALIZZAZIONE}(s)$

$$M_1 = \{m \mid \text{medico}(m) \wedge \text{PRIMARIA}(m, s)\}$$

$$M_2 = \{m \mid \text{medico}(m) \wedge \text{SECONDARIA}(m, s)\}$$

Result è tale che

$$|M_1| > 0 \rightarrow \text{Result} = M_1$$

$$\wedge |M_1| = 0 \rightarrow \text{Result} = M_2$$

$\text{visite} (m: \text{Medico}): \text{Stanza} [0..*]$

Pre:

Post:

$$S = \left\{ s \mid \text{STANZA}(s) \wedge \left( \exists r \text{ Ricovero}(r) \wedge \text{Sta\_RIC}(s, r) \wedge \text{med\_ric}(m, r) \right) \right\}$$

$$S_{\text{sorted}} = \text{sorted}(S, \text{sortBy PianoSettore})$$

$$\text{Result} = S_{\text{sorted}}$$

$\text{sortByPianoSettore} (s: \text{Stanza}, s': \text{Stanza}): \text{Booleano}$

• PRE:

• POST:

Siano  $p, p', se, se'$  tali che

$$\text{Piano}(s, p) \wedge \text{Piano}(s', p') \wedge \text{Settore}(s, se) \\ \wedge \text{Settore}(s', se')$$

$$\left( \left( p < p' \vee (p = p' \wedge se \leq se') \right) \rightarrow \text{result} = \text{True} \right) \\ \wedge \\ \left( \left( p > p' \vee (p = p' \wedge se > se') \right) \rightarrow \text{Result} = \text{False} \right)$$

Risposta alla Domanda 5 (segue)

## 2 Progettazione della base dati e delle funzionalità

**Domanda 6 (20 minuti; 30 minuti al massimo)** Iniziare la fase di progettazione logica della base di dati decidendo il DBMS da utilizzare e ristrutturando lo schema UML delle classi concettuale, il dizionario dei dati e i vincoli esterni. In particolare:

- OK • progettare una corrispondenza tra i tipi di dato concettuali ed opportuni domini SQL (domini base o utente, oppure realizzati mediante relazioni aggiuntive) supportati dal DBMS scelto
- OK • eliminare attributi multivalore o composti
- OK • eliminare relazioni is-a e generalizzazioni
- OK • definire un identificatore primario per ogni classe
- OK • ristrutturare i vincoli esterni per renderli consistenti con la struttura del nuovo diagramma.

Descrivere brevemente le principali scelte effettuate.

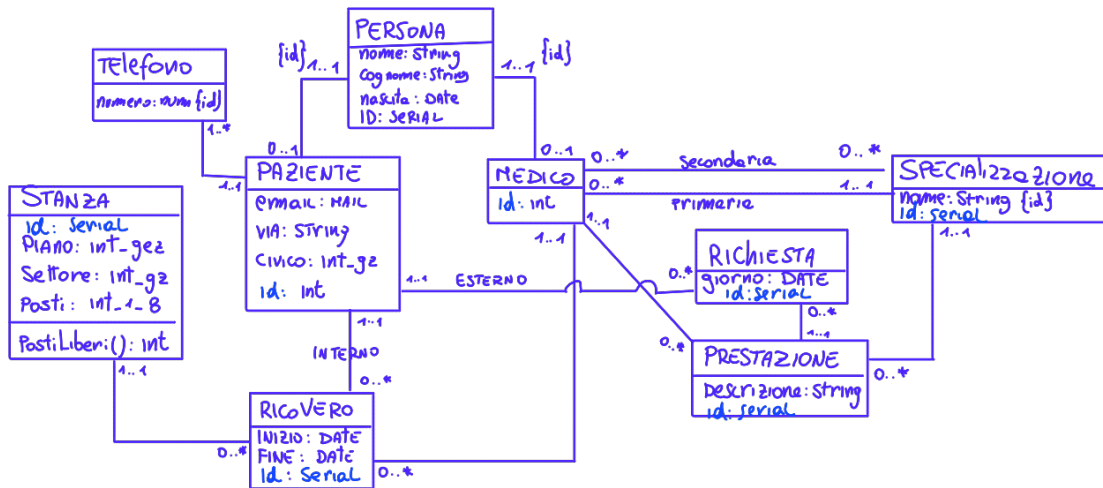
DBMS da utilizzare ..... PostgreSQL

Corrispondenza tra tipi di dato concettuali e domini supportati dal DBMS

```
Create domain String AS VARCHAR
Create domain int_gz AS INT check(value > 0)
Create domain int_gez AS INT check(value >= 0)
CREATE domain int_1_8 AS INT check(value > 0 AND value < 9)
Create domain MAIL AS VARCHAR CHECK(value ~ '...')
Create domain NUM AS VARCHAR check(value ~ '..')
```



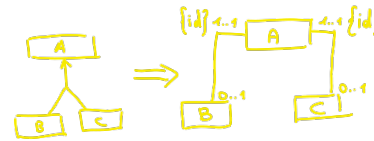
## Diagramma UML delle classi ristrutturato



FINE PUO ESSERE NULLO

**Breve descrizione delle scelte effettuate durante la ristrutturazione**Rimozione Attributo multivalore telefono  $\rightarrow$  NUOVA CLASSERimozione Attributo composto indirizzo  $\rightarrow$  diviso in 2 attributiRimozione IS-a PERSONA  $\rightarrow$  Sostituzione

Rimozione IS-a ASSOCIAZIONE PRIMARIA/secondaria

**Vincoli esterni introdotti o modificati durante la fase di ristrutturazione**

(si omettano i vincoli esterni la cui formulazione è rimasta identica a seguito della ristrutturazione)

[V. Complete su Paziente]

 $\forall p \text{ Paziente}(p) \rightarrow \exists i \text{ Paz\_int}(p, i) \vee \exists e \text{ Paz\_est}(p, e)$ 

[V. Complete su Persona]

 $\forall p \text{ Persona}(p) \rightarrow \exists_{pa} \text{ Per\_paz}(p, pa) \vee \exists_m \text{ Per\_med}(p, m)$ 

② [V. non si può essere medico di se stesso]

$$\forall pa, pe, m, r \quad \text{Persona}(pe) \wedge \text{Per\_paz}(pe, pa) \wedge \text{Per\_med}(pe, m) \wedge \text{Interno}(pa, r) \\ \rightarrow \neg \text{med\_ric}(m, r)$$

## Risposta alla Domanda 6 (segue)

Ricorda:

È possibile procedere per accorpamenti ogni volta  
che un'Associazione ha un ruolo a molti 1..1 o 0..1

**Domanda 7 (30 minuti; 60 minuti al massimo)** Proseguire la fase di progettazione logica della base di dati producendo lo schema relazionale della base dati e i relativi vincoli a partire dallo schema UML delle classi ristrutturato.

Una risposta soddisfacente a questa domanda è condizione *necessaria* (ma non sufficiente) per superare la prova.

FK = Foreign Key    REF = References

1 Relazione TELEFONO (nome) Derivante da: classe | associazione (cerchiare)

Attributi | numero | PAZIENTE | | | | | |

Domini | num | int | | | | | |

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con \*

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

FK PAZIENTE REF PAZIENTE(ID);

La relazione accorpa le relazioni che implementano le seguenti associazioni: TEL - PAZ .....

2 Relazione PERSONA (nome) Derivante da: classe | associazione (cerchiare)

Attributi | ID | Nome | Cognome | Nascita | | | |

Domini | Serial | String | String | Date | | | |

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con \*

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

CHECK (nascita ≤ current\_date)

La relazione accorpa le relazioni che implementano le seguenti associazioni: .....

3 Relazione PAZIENTE (nome) Derivante da: classe | associazione (cerchiare)

Attributi | PERSONA | Via | Civico | email | | | |

Domini | int | String | int\_gz | MAIL | | | |

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con \*

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

FK PERSONA REF PERSONA(ID);

La relazione accorpa le relazioni che implementano le seguenti associazioni: PAZ - PER .....

4 Relazione MEDICO (nome) Derivante da: classe | associazione (cerchiare)

Attributi | PERSONA | SPEC-PR | | | | | |

Domini | int | int | | | | | |

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con \*

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

FK PERSONA REF PERSONA(ID);    FK SPEC-PR REF Specializzazione(ID)

Unique(PERSONA);

La relazione accorpa le relazioni che implementano le seguenti associazioni: MED - PER, primaria .....

5 Relazione RICOVERO (nome) Derivante da: classe | associazione (cerchiare)

Attributi | ID | INIZIO | FINE \* | Stanze | PAZIENTE | Medico | |

Domini | Serial | DATE | DATE | int | int | int | |

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con \*

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

FK STANZA REF STANZA(ID);    CHECK (INIZIO ≤ FINE);

FK PAZIENTE REF PAZIENTE(ID);

FK Medico REF Medico(ID);

CHECK (PAZIENTE <> Medico);

La relazione accorpa le relazioni che implementano le seguenti associazioni: RIC\_STA, INTERNO, RIC\_MED .....

6 Relazione STANZA ..... (nome) Derivante da: classe | associazione (cerchiare)

Attributi	<u>ID</u>	<u>PIANO</u>	<u>Settore</u>	<u>Posti</u>				
Domini	<u>Serial</u>	<u>int-gez</u>	<u>int-gez</u>	<u>int-1-8</u>				

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con \*

Vincoli (foreign key, inclusione, altra chiave, di ennuola, di dominio):

La relazione accorpa le relazioni che implementano le seguenti associazioni: .....

7 Relazione PRESTAZIONE ..... (nome) Derivante da: classe | associazione (cerchiare)

Attributi	<u>ID</u>	<u>DESCRIZIONE</u>	<u>Medico</u>	<u>SPEC</u>				
Domini	<u>Serial</u>	<u>String</u>	<u>int</u>	<u>int</u>				

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con \*

Vincoli (foreign key, inclusione, altra chiave, di ennuola, di dominio):

La relazione accorpa le relazioni che implementano le seguenti associazioni: med\_PRE, PRE\_SPE .....

8 Relazione RICHIESTA ..... (nome) Derivante da: classe | associazione (cerchiare)

Attributi	<u>ID</u>	<u>giorno</u>	<u>PAZIENTE</u>	<u>PRESTAZIONE</u>				
Domini	<u>Serial</u>	<u>date</u>	<u>int</u>	<u>int</u>				

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con \*

Vincoli (foreign key, inclusione, altra chiave, di ennuola, di dominio):

check(current\_date < giorno);

La relazione accorpa le relazioni che implementano le seguenti associazioni: RIC\_PRE, ESTERNO .....

9 Relazione Specializzazione ..... (nome) Derivante da: classe | associazione (cerchiare)

Attributi	<u>id</u>	<u>nome</u>						
Domini	<u>Serial</u>	<u>String</u>						

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con \*

Vincoli (foreign key, inclusione, altra chiave, di ennuola, di dominio):

Unique(nome);

La relazione accorpa le relazioni che implementano le seguenti associazioni: .....

10 Relazione Secondaria ..... (nome) Derivante da: classe | associazione (cerchiare)

Attributi	<u>Medico</u>	<u>SPEC</u>						
Domini	<u>int</u>	<u>int</u>						

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con \*

Vincoli (foreign key, inclusione, altra chiave, di ennuola, di dominio):

La relazione accorpa le relazioni che implementano le seguenti associazioni: .....

**11 Relazione** ..... (nome) Derivante da: **classe** | **associazione** (cerchiare)

Attributi								
Domini								

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con \*

Vincoli (foreign key, inclusione, altra chiave, di enupla, di dominio):

La relazione accorpa le relazioni che implementano le seguenti associazioni: .....

**12 Relazione** ..... (nome) Derivante da: **classe** | **associazione** (cerchiare)

Attributi								
Domini								

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con \*

Vincoli (foreign key, inclusione, altra chiave, di enupla, di dominio):

La relazione accorpa le relazioni che implementano le seguenti associazioni: .....

**13 Relazione** ..... (nome) Derivante da: **classe** | **associazione** (cerchiare)

Attributi								
Domini								

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con \*

Vincoli (foreign key, inclusione, altra chiave, di enupla, di dominio):

La relazione accorpa le relazioni che implementano le seguenti associazioni: .....

**14 Relazione** ..... (nome) Derivante da: **classe** | **associazione** (cerchiare)

Attributi								
Domini								

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con \*

Vincoli (foreign key, inclusione, altra chiave, di enupla, di dominio):

La relazione accorpa le relazioni che implementano le seguenti associazioni: .....

**15 Relazione** ..... (nome) Derivante da: **classe** | **associazione** (cerchiare)

Attributi								
Domini								

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con \*

Vincoli (foreign key, inclusione, altra chiave, di enupla, di dominio):

La relazione accorpa le relazioni che implementano le seguenti associazioni: .....

**16 Relazione** ..... (nome) Derivante da: **classe** | **associazione** (cerchiare)

Attributi | | | | | | | |

Domini | | | | | | | |

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con \*

Vincoli (foreign key, inclusione, altra chiave, di ennuola, di dominio):

La relazione accorpa le relazioni che implementano le seguenti associazioni: .....

**17 Relazione** ..... (nome) Derivante da: **classe** | **associazione** (cerchiare)

Attributi | | | | | | | |

Domini | | | | | | | |

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con \*

Vincoli (foreign key, inclusione, altra chiave, di ennuola, di dominio):

La relazione accorpa le relazioni che implementano le seguenti associazioni: .....

**18 Relazione** ..... (nome) Derivante da: **classe** | **associazione** (cerchiare)

Attributi | | | | | | | |

Domini | | | | | | | |

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con \*

Vincoli (foreign key, inclusione, altra chiave, di ennuola, di dominio):

La relazione accorpa le relazioni che implementano le seguenti associazioni: .....

**19 Relazione** ..... (nome) Derivante da: **classe** | **associazione** (cerchiare)

Attributi | | | | | | | |

Domini | | | | | | | |

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con \*

Vincoli (foreign key, inclusione, altra chiave, di ennuola, di dominio):

La relazione accorpa le relazioni che implementano le seguenti associazioni: .....

**20 Relazione** ..... (nome) Derivante da: **classe** | **associazione** (cerchiare)

Attributi | | | | | | | |

Domini | | | | | | | |

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con \*

Vincoli (foreign key, inclusione, altra chiave, di ennuola, di dominio):

La relazione accorpa le relazioni che implementano le seguenti associazioni: .....

**Ulteriori vincoli esterni**

Per ogni ulteriore vincolo esterno (non ancora espresso perché non definibile mediante vincoli di chiave, foreign key, enunpla, dominio, inclusione), progettare un trigger che lo implementi, definendo: (a) gli eventi da intercettare (inserimento, modifica, eliminazione di enunple); (b) quando intercettare tali eventi (appena prima o subito dopo l'evento intercettato); (c) la relativa funzione in pseudo-codice con SQL immerso che implementa il controllo del vincolo.

[T. Impedisci A medici non competenti di svolgere le prestazioni]

```

OK = EXISTS (
    Select * FROM Medico m, Secondaria s
    WHERE NEW.medico = m.PERSONA AND ((m.SPEC = NEW.SPEC) OR (m.SPEC = S.MEDICO AND S.SPE = NEW.SPEC))
)
IF OK = 'TRUE'
    Permetti op.
ELSE
    genera Err.
  
```

[T. No overlaps ricoveri]

Insert o Update  
Prima dell'op

```

ERROR = EXISTS (
    Select * FROM Ricoveri
    Where (NEW.INIZIO, NEW.FINE) OVERLAPS (INIZIO, FINE) AND NEW.PAZIENTE = PAZIENTE
)
IF ERROR = 'TRUE'
    genera ERR.
ELSE
    Permetti op
  
```

[T. Numero Ricoveri minore dei Posti]

Insert o Update  
Prima dell'op

```

IF (Select count(*) FROM Ricoveri r WHERE STANZA = NEW.STANZA) >= (Select Posti FROM STANZA WHERE Id = NEW.STANZA)
    genera Err.
else
    Permetti op
  
```



Risposta alla **Domanda 7** (segue)

**Domanda 8 (30 minuti; 45 minuti al massimo)** Proseguire la fase di progettazione dell'applicazione producendo le specifiche realizzative delle operazioni di classe e/o use-case definite per modellare i requisiti contrassegnati dalla barra laterale della specifica dei requisiti.

In particolare, per ogni operazione definire la segnatura, in termini di nome dell'operazione, nomi e dominio SQL degli argomenti, dominio SQL dell'eventuale valore di ritorno, e un algoritmo in pseudo-codice con SQL immerso che verifichi le precondizioni e garantisca il raggiungimento delle postcondizioni definite in fase di Analisi. Specificare, per ogni operazione, se debba essere implementata nel DBMS o nel *back-end*.

Una risposta soddisfacente a questa domanda è condizione *necessaria* (ma non sufficiente) per superare la prova.

Risposta Use-CASE

*mediciIdonei*(*p: Integer*): *INSIEME*(<*Integer*>)

```
P = Select m.PERSONA FROM MEDICI AS m
      WHERE P.SPEC = m.SPEC
```

```
S = Select s.medico FROM Secondaria AS s
      WHERE P.SPEC = s.ID
```

```
IF P != NULL
  RETURN P
```

```
ELSE
  RETURN S
```

*Visite*(*m: Integer*): *LISTA*(<*Integer*>)

```
Select s.ID from STANZA s, Ricovero r
      WHERE m.PERSONA = r.medico AND r.STANZA = s.ID
      order by s.Piano, s.settore;
```

OPERAZIONE di classe

*PostiLiberi*(*id-STANZA: int*):<*Integer*>

```
OCCUPATI = Select count(*) FROM RICOVERI AS r
      WHERE id-STANZA = r.STANZA
```

```
TOT = Select posti FROM STANZA where id-STANZA = id
```

```
RETURN TOT - OCCUPATI
```

Risposta alla **Domanda 8** (segue)

Tempo totale stimato per svolgere questa prova: 180 minuti (tempo totale concesso: 300 minuti).  
[Spazio per minute. Questa pagina non sarà valutata a meno che non sia puntata da pagine precedenti.]

[Spazio per minute. Questa pagina non sarà valutata a meno che non sia puntata da pagine precedenti.]





Matricola: .....

*Minute*

[Spazio per minute. Questa pagina non sarà valutata a meno che non sia puntata da pagine precedenti.]

[Spazio per minute. Questa pagina non sarà valutata a meno che non sia puntata da pagine precedenti.]

