



Sapienza Università di Roma
Facoltà di Ing. dell'Informazione, Informatica e Statistica, Laurea in Informatica
Insegnamento di **Basi di Dati, Modulo 2**
Prof. Toni Mancini
Dipartimento di Informatica
<http://tmancini.di.uniroma1.it>

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:

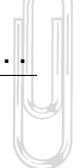
- Laurea in Informatica, canale 1 (Prof. G. Perelli)
- Laurea in Informatica, canale 2 (Prof.ssa M. De Marsico)

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), *inclusa* 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)

Raffinare le specifiche

PAZIENTI (int)

- nome
- Cognome
- Nascita
- telefono
- email
- Postale

PAZ. (EST)

- nome
- Cognome
- Nascita
- Richieste prestazioni con DATA

Medici

- nome
- Cognome
- Nascita
- PAZIENTI (0..N)
- Specializzazione primaria (1..1)
- Spec. secondaria (0..N)

Ricovero

- DATA Inizio
- PAZiente
- Solo se c'è un posto letto disponibile
- DATA Fine
- Posto letto

STANZA

- Letti (1..8)
- Piano (int pos)
- Lettore (int pos)

PRESTAZIONI

- V. PAZIENTI ESTERNI
- Specializzazione
- Descrizione

Specializzazione

- TIPO

funzionalità:

+ Registrazione del ricovero (sia inizio che fine)
[Personale di accettazione]

+ Calcolare l'itinerario di un medico (un insieme ordinato delle STANZE DEI PAZIENTI che ha in CURA)
[medico]

→ ordinamento:
Piano della STANZA (cresc.)
↓
Settore della STANZA (cresc.)

+ DATA Una Richiesta da PAZIENTE esterno
per una specializzazione. Bisogna restituire
l'insieme dei medici

↳ Se esiste medici con spec. primaria
Ritorna questo insieme

↳ Altrimenti ritorna medici con spec.
secondaria

Il Sistema è Accessibile

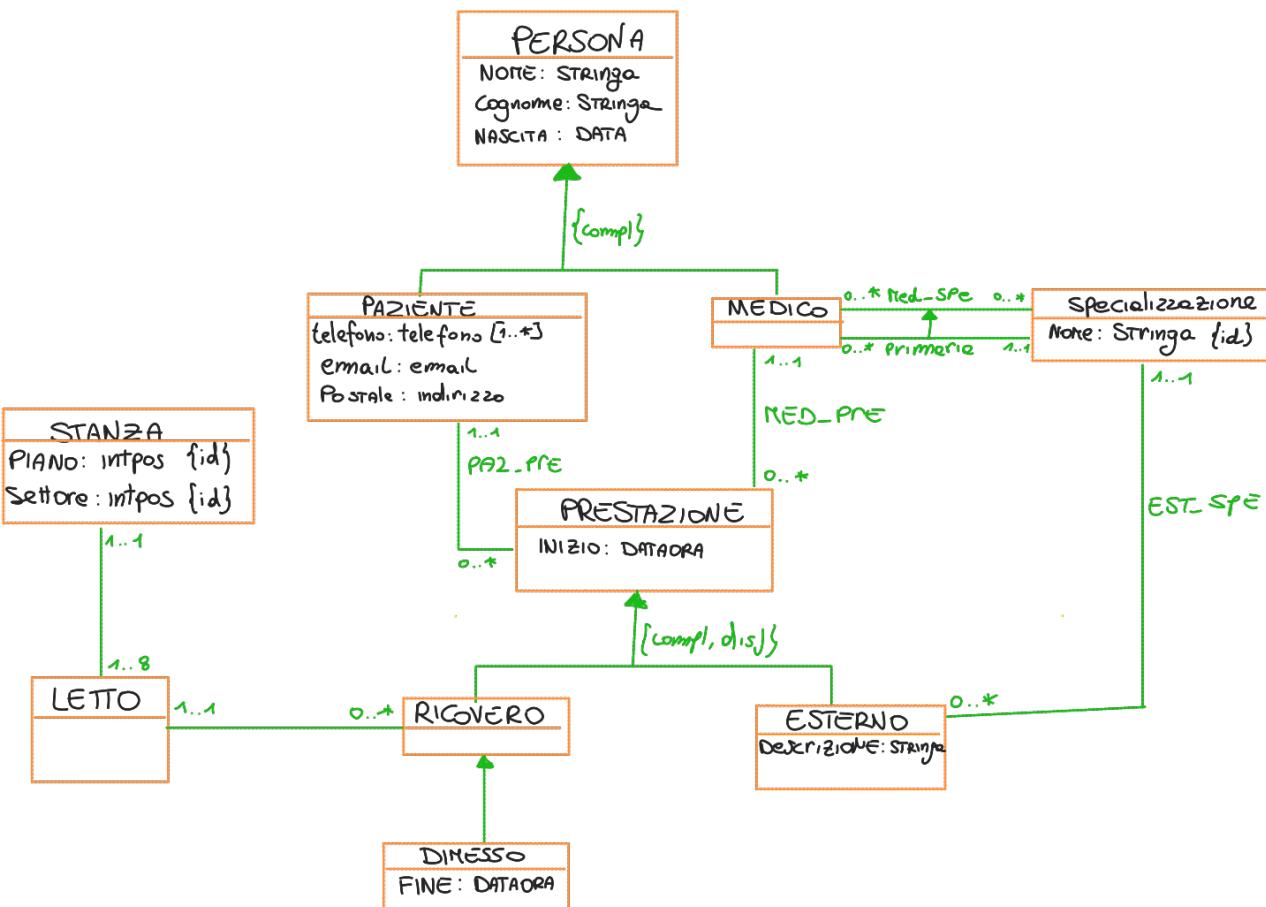
Ai medici, Personale Amministrativo
e ufficio prenotazioni

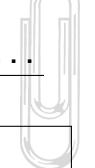
Domanda 2 (45 minuti; 75 minuti al massimo) Proseguire la fase di Analisi Concettuale dei requisiti, producendo un diagramma UML concettuale delle classi per l'applicazione, le specifiche di classi, associazioni, tipi di dato e vincoli esterni.

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 classe 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 assoc .
Semantica di $\text{assoc}(c_1, c_2)$: (c_1, c_2) è una istanza di assoc .
 - Un simbolo di predicato $\text{attr}/2$ per ogni attributo attr di entità.
Semantica di $\text{attr}(c, v)$: uno dei valori dell'attributo attr dell'istanza c è v .
 - Un simbolo di predicato $\text{attr}/3$ per ogni attributo attr di associazione binaria.
Semantica di $\text{attr}(c_1, c_2, v)$: uno dei valori dell'attrib. 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 $\text{op}(c, \text{arg}_1, \dots, \text{arg}_n, v)$: uno dei valori di ritorno di op , quando invocata sull'istanza c e con argomenti $\text{arg}_1, \dots, \text{arg}_n$ 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><input type="checkbox"/> Tipo: Classe Associazione (cerchiare)</p> <p>Nome: <u>DIMESSO</u></p> <p>Operazioni, vincoli: $\neg \exists d, i, f \ (Dimesso(d) \wedge \text{INIZIO}(d, i) \wedge \text{FINE}(d, f) \rightarrow f > i)$ $\neg \exists d, t \ (dimesso(d) \wedge \text{fine}(d, t) \rightarrow t \leq \text{adesso})$ $\forall r, r', l, i, \ddot{o} \ (\text{RICOVERO}(r) \wedge \text{RICOVERO}(r') \wedge \text{let_RIC}(l, r) \wedge \text{let_RIC}(l, r') \wedge \text{INIZIO}(r, i) \wedge \text{INIZIO}(r', i) \rightarrow \exists t \ \text{dataora}(t) \wedge (i \leq t \wedge (\forall f \ \text{fine}(r, f) \rightarrow t \leq f)) \wedge (i' \leq t \wedge (\forall f' \ \text{FINE}(r', f') \rightarrow t \leq f'))$</p>	<p><input type="checkbox"/> Tipo: Classe Associazione (cerchiare)</p> <p>Nome: <u>RICOVERO</u></p> <p>Operazioni, vincoli: $\forall r, r', l, i, \ddot{o} \ (\text{RICOVERO}(r) \wedge \text{RICOVERO}(r') \wedge \text{let_RIC}(l, r) \wedge \text{let_RIC}(l, r') \wedge \text{INIZIO}(r, i) \wedge \text{INIZIO}(r', i) \rightarrow \exists t \ \text{dataora}(t) \wedge (i \leq t \wedge (\forall f \ \text{fine}(r, f) \rightarrow t \leq f)) \wedge (i' \leq t \wedge (\forall f' \ \text{FINE}(r', f') \rightarrow t \leq f'))$</p>
--	---

3	Tipo: Classe Associazione (cerchiare)
	<p>Nome: .. MEDICO ..</p> <p>Operazioni, vincoli:</p> <p>Vincolo: un medico che è anche paziente non può essere medico di se stesso</p> <p>$\nexists_{m,p}$ $(medico(m) \wedge paziente(m) \wedge PAZ_PRE(m,p) \wedge PRESTAZIONE(p)) \longrightarrow \neg MED_PRE(m,p)$</p> <p>Vincolo: un medico non può avere prestazioni esterne se non ha la specializzazione adatta.</p> <p>$\nexists_{m,p,s}$ $(medico(m) \wedge ESTERNO(p) \wedge SPECIALIZZAZIONE(s) \wedge EST_SPEC(p,s) \wedge \neg MED_SPEC(m,s)) \longrightarrow \neg MED_PRE(m,p)$</p>

6	Tipo: Classe Associazione (cerchiare)
	<p>Nome: .. PAZIENTE ..</p> <p>Operazioni, vincoli:</p> <p>Vincolo: un paziente non può essere ricoverato più volte contemporaneamente</p> <p>$\nexists_{p,r,r',i,i'}$ $(PAZIENTE(p) \wedge Ricovero(r) \wedge Ricovero(r') \wedge INIZIO(r,i) \wedge r \neq r' \wedge INIZIO(r',i') \wedge PAZ_PRE(p,r) \wedge PAZ_PRE(p,r')) \longrightarrow \neg \exists d \text{ data} (d) \wedge (i \leq d \wedge (\nexists f \text{ fine}(r,f) \rightarrow t \leq f) \wedge (i' \leq d \wedge (\nexists f' \text{ fine}(r',f') \rightarrow t \leq f'))$</p>

4	Tipo: Classe Associazione (cerchiare)
	<p>Nome: .. PERSONA ..</p> <p>Operazioni, vincoli:</p> <p>V. LE NASCITE delle persone non possono essere nel futuro</p> <p>$\nexists_{p,n}$ $(PERSONA(p) \wedge NASCITA(p,n) \longrightarrow n \leq \text{adesso})$</p>

7	Tipo: Classe Associazione (cerchiare)
	<p>Nome: .. PRESTAZIONE ..</p> <p>Operazioni, vincoli:</p> <p>Vincolo: ogni data di inizio deve essere passata</p> <p>$\nexists_{p,d}$ $(PRESTAZIONE(p) \wedge INIZIO(p,d) \longrightarrow d \leq \text{adesso})$</p>

5	Tipo: Classe Associazione (cerchiare)
	<p>Nome: ..</p> <p>Operazioni, vincoli:</p>

8	Tipo: Classe Associazione (cerchiare)
	<p>Nome: ..</p> <p>Operazioni, vincoli:</p>

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

SPECIFICHE DEI DATI

Stringa : Stringe secondo STANDARD

intpos : intero > 0

telefono : Stringe secondo STANDARD

DATA : Secondo STANDARD es. 0000/mmm/99

DATAORA : Secondo STANDARD es. 0000/mmm/99/11/min/Sec

EMAIL : Stringe secondo STANDARD

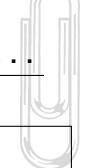
indirizzo : (via: Stringe, civico : Stringe)

OVERLAPS(i : DATAORA, f : DATAORA, i' : DATAORA, f' : DATAORA) : Booleano

PRE-CONDIZIONE : $i \leq f \wedge i' \leq f'$

POST-CONDIZIONE :

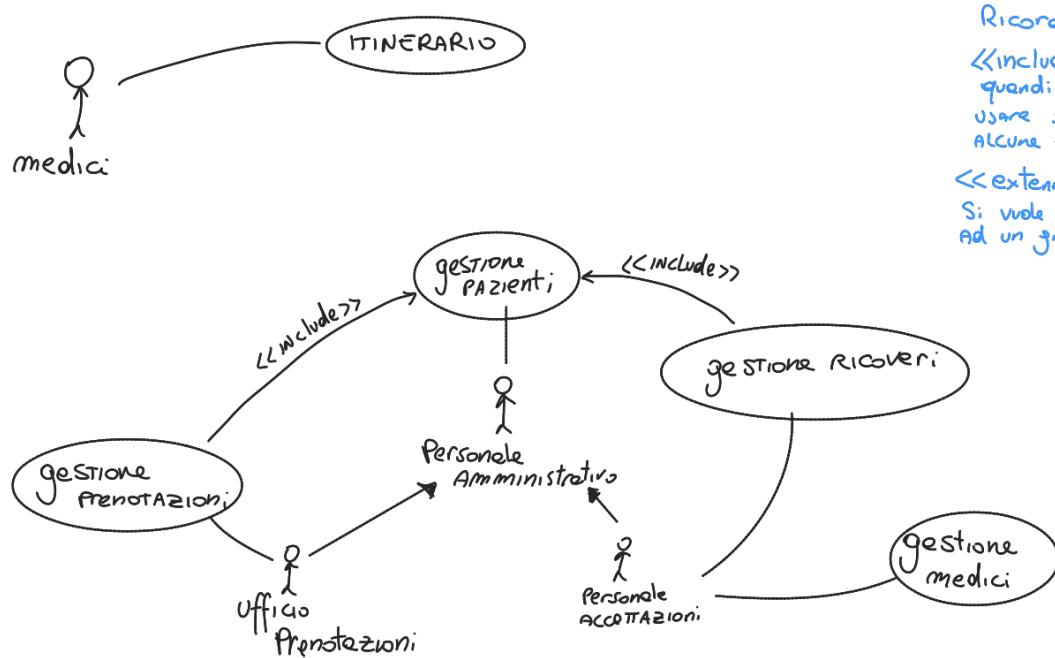
RESULT TRUE $\longleftrightarrow (\exists t \text{ DATAORA}(t) \wedge i \leq t \leq f \wedge i' \leq t \leq f')$



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



Ricorda

«include» si usa quando si vuole far usare solo una parte di alcune funzionalità

«extend» Si usa quando si vuole dare altre funzionalità ad un gruppo di funzionalità



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

■ ITINERARIO

Def MostraItinerario(m: Medico) : STANZA [0..*]

■ Gestione Ricoveri

AggiungiRicovero(p: PAZIENTE, m: medico, i: DATAORA) : Ricovero

DimettiPaziente(r: Ricovero) : Dimesso

Def PostiDisponibili() : intero ≥ 0

■ GESTIONE MEDICI

Def CalcolaMediciCompetenti(s: specializzazione) : Medico [0..*]

■ Gestione PAZIENTI

RegistraPaziente(n: stringa, c: stringa, d: data, t: telefono[0..*], e: email, i: indirizzo) : PAZIENTE

■ GESTIONE PRENOTAZIONI

PrenotazioneEsterna(p: PAZIENTE, dt: DATAORA, s: specializzazione) : ESTERNO



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

CALCOLA MEDICO COMPETENTI (*s: specializzazione*) : Medico [0..*]

Pre-condizione: nessuna

Post-condizione:

$$P = \{m \mid \text{medico}(m) \wedge \text{primaria}(m, s)\}$$

$$S = \{m \mid \text{medico}(m) \wedge \text{med-spe}(m, s)\}$$

$$(P \neq \emptyset \rightarrow \text{RESULT} = P) \wedge (P = \emptyset \rightarrow \text{RESULT} = S)$$

POST DISponibili: () : intero ≥ 0

Pre: nessuna

Post:

$$L = \left| \{l \mid \text{Letto}(l) \wedge (\neg \exists r \text{ ricovero}(r) \wedge \text{LET-RIC}(l, r) \wedge \neg \text{Dimesso}(r))\} \right|$$

$$\text{RESULT} = L$$

MOSTRA ITINERARIO (*m: Medico*) : STANZA [0..*]

Pre-Condizione: NESSUNA

Post-Condizione:

$$S = \left\{ s \mid \text{stanza}(s) \wedge \left(\exists l, r \text{ letto}(l) \wedge \text{ricovero}(r) \wedge \text{LET-RIC}(l, r) \wedge \text{MED-PRE}(m, r) \wedge \neg \text{DIMESSO}(r) \right) \right\}$$

$$S_o = \text{Sorted}(S, \text{SortbyPianoSettore})$$

$$\text{RESULT} = S_o$$

SortbyPianoSettore (*s: STANZA, s': STANZA*) : Booleano

Pre: nessuna

Post:

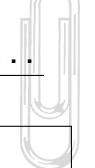
$$\forall p_i, p'_i, st, st'$$

$$(\text{PIANO}(s, p_i) \wedge \text{PIANO}(s', p'_i) \wedge \text{settore}(s, st) \wedge \text{settore}(s', st'))$$

$$\wedge ((p_i < p'_i \vee (p_i = p'_i \wedge st < st')) \rightarrow \text{RESULT} = \text{TRUE})$$

\wedge

$$(\neg (p_i < p'_i \vee (p_i = p'_i \wedge st < st')) \rightarrow \text{RESULT} = \text{FALSE})$$



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:

- 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
- eliminare attributi multivale o composti
- eliminare relazioni is-a e generalizzazioni
- definire un identificatore primario per ogni classe
- ristrutturare i vincoli esterni per renderli consistenti con la struttura del nuovo diagramma.

Descrivere brevemente le principali scelte effettuate.

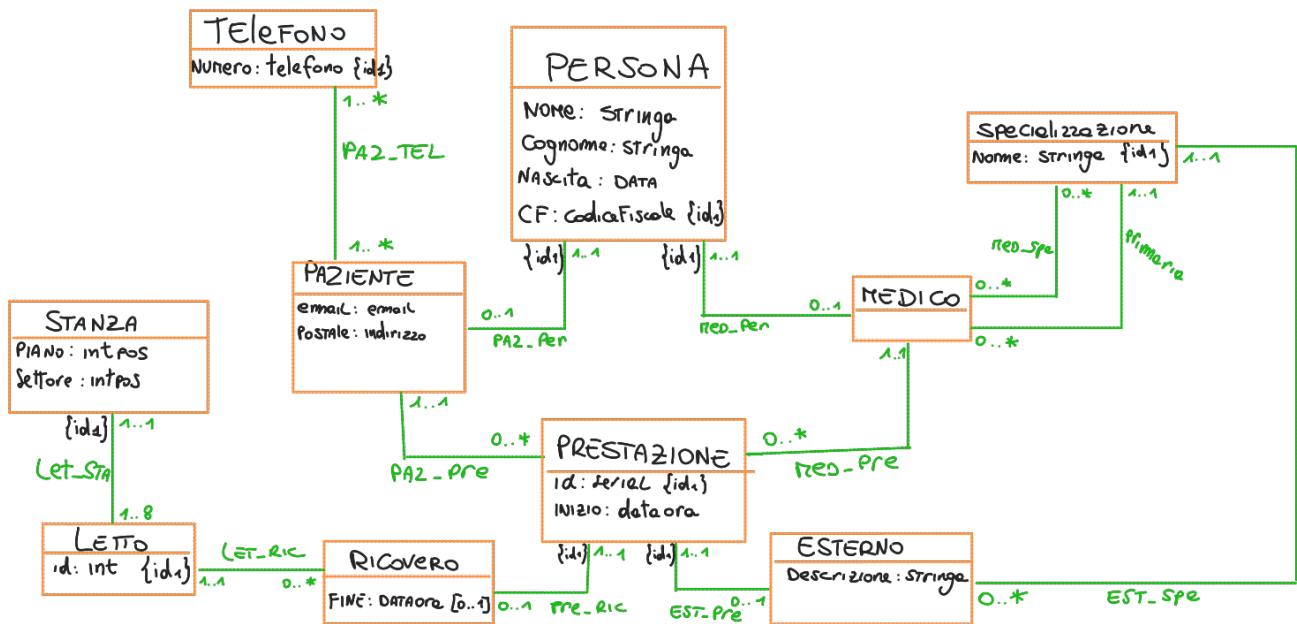
DBMS da utilizzare ... **POSTGRESQL**.....
Corrispondenza tra tipi di dato concettuali e domini supportati dal DBMS

```

Create Domain Stringa AS TEXT
Create Domain intpos AS INT check(VALUE > 0)
Create Domain telefono AS TEXT check(VALUE ~ '...') Regex
CREATE DOMAIN DATA AS DATE
CREATE DOMAIN DATAORA AS TIMESTAMP
Create Domain email AS TEXT check(VALUE ~ '... ')
Create type INDIRIZZO AS (
    VIA STRINGA,
    CIVICO CIVICO.VALIDO
)
Create Domain civico_valido AS TEXT CHECK(VALUE ~ '... ')
Create Domain codiceFiscale AS TEXT check(VALUE ~ '... ')

```

Diagramma UML delle classi ristrutturato



Breve descrizione delle scelte effettuate durante la ristrutturazione

Rimosso attributo multivalore telefono creando classe apposita

IS-A Medico/paziente rimossa per sostituzione

IS-A Ricovero/esterno rimossa per sostituzione

IS-A Ditta rimossa per fusione

IS-A tra relazioni: Specializzazioni secondarie e primarie

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)

Vincolo: IS-A Medico/paziente complete (new)

$$\nexists p \text{ (PERSONE}(p) \longrightarrow [\exists m \text{ MED_PER}(m,p)] \vee [\exists pa \text{ PAZ_PER}(pa,p)])$$

Vincolo: IS-A Ricovero/esterno complete+disjunt (new)

$$\nexists p \text{ (PRESTAZIONE}(p) \rightarrow [\exists r \text{ PRE_RIC}(p,r) \wedge \neg \exists e \text{ PRE_RIC}(p,e)] \vee [\exists e \text{ EST_R}(e,p) \wedge \neg \exists r \text{ PRE_RIC}(e,r)])$$

① Vincolo: la data di fine del ricovero e' > dell'inizio (mod)

$$\nexists r, f, i, p \text{ (Ricovero}(r) \wedge \text{fine}(r,f) \wedge \text{Prestazione}(p) \wedge \text{pre_RIC}(p,r) \wedge \text{INIZIO}(r,i) \longrightarrow f > i)$$

Vincolo: la data di dimissione e' ≤ adesso

$$\nexists r, f \text{ (Ricovero}(r) \wedge \text{fine}(r,f) \longrightarrow f \leq \text{adesso})$$

② Vincolo: un letto non puo' avere piu' ricoveri sovrapposti (mod)

$$\nexists l, r, p, i, r', p', i'$$

$$(\text{Letto}(l) \wedge \text{Ricovero}(r) \wedge \text{let_RIC}(l,r) \wedge \text{pre_RIC}(p,r) \wedge \text{PRESTAZIONE}(p) \wedge \text{INIZIO}(p,i) \wedge \text{Ricovero}(r') \wedge \text{let_RIC}(l,r') \wedge \text{pre_RIC}(p',r'))$$

$$\text{prestazione}(p') \wedge r \neq r' \wedge p \neq p' \wedge \text{INIZIO}(p',i') \longrightarrow \neg \exists t \text{ dataora}(t) \wedge (i \leq t \wedge (\nexists f \text{ fine}(r,f) \rightarrow t \leq f)) \wedge (i' \leq t \wedge (\nexists f' \text{ fine}(r',f') \rightarrow t \leq f'))$$

③ Vincolo: Se e' primaria allora non esiste quella secondaria

$$\nexists m, s$$

$$(\text{medico}(m) \wedge \text{specializzazione}(s) \wedge \text{primaria}(m,s) \longrightarrow \text{med_spe}(m,s))$$

Vincolo: un medico non puo' essere medico di se stesso

$$\nexists m, p, pa, pre$$

$$(\text{medico}(m) \wedge \text{paziente}(pa) \wedge \text{persona}(p) \wedge \text{PAZ_PER}(pa,p) \wedge \text{med_per}(m,p) \wedge \text{prestazione}(pre) \wedge \text{PAZ_PRE}(p,pre) \longrightarrow \neg \text{med_pre}(m,pre))$$

Risposta alla Domanda 6 (segue)

① **Vincolo:** NO prestazioni esterne SENZA specializzazione corrette

$\forall m, p, e, s$

(Medico(m) \wedge prestazione(p) \wedge esterno(e) \wedge med_pre(m, p) \wedge est_pre(e, p) \wedge specializzazioni(s) \wedge est_spe(e, s))
 \rightarrow med_spe(m, s))

② **Vincolo:** NO ricoveri di un paziente contemporaneamente

$\forall p, pr, i, pr', r, r', i, i'$

(PAZIENTE(p) \wedge prestazione(pr) \wedge prestazione(pr') \wedge PAZ_PRE(p, pr) \wedge PAZ_PRE(p', pr') \wedge Ricovero(r) \wedge Ricovero(r') \wedge PRE_RIC(pr, r)
 \wedge PRE_RIC(pr', r') \wedge INIZIO(pr, i) \wedge INIZIO(pr', i') \wedge pr \neq pr' \wedge r \neq r'
 $\rightarrow \neg \exists t \text{ dataora}(t) \wedge (i \leq t \wedge (\text{ff FINE}(r, f) \rightarrow t \leq f)) \wedge$
 $(i' \leq t \wedge (\text{ff' FINE}(r', f') \rightarrow t \leq f'))$

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.

[1] Relazione .TELEFONO... (nome)	Derivante da: classe associazione (cerchiare)
--	---

Attributi <u>NUMERO</u>						
---------------------------	--	--	--	--	--	--

Domini <u>telefono</u>						
--------------------------	--	--	--	--	--	--

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):

Numero \subseteq PAZ-TEL(telefono)

La relazione accorda le relazioni che implementano le seguenti associazioni:

[2] Relazione ..PAZ-TEL... (nome)	Derivante da: classe associazione (cerchiare)
--	---

Attributi <u>telefono</u> <u>PAZiente</u>						
---	--	--	--	--	--	--

Domini <u>telefono</u> 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 Telefono REF Telefono(Numero)

FK PAZiente REF PAZiente (id)

La relazione accorda le relazioni che implementano le seguenti associazioni:

[3] Relazione PAZIENTE... (nome)	Derivante da: classe associazione (cerchiare)
---	---

Attributi <u>email</u> <u>POSTALE</u> <u>PERSONA</u>						
--	--	--	--	--	--	--

Domini <u>email</u> <u>indirizzo</u> <u>CodiceFiscale</u>						
---	--	--	--	--	--	--

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(CF)

La relazione accorda le relazioni che implementano le seguenti associazioni:

[4] Relazione ..PERSONA... (nome)	Derivante da: classe associazione (cerchiare)
--	---

Attributi <u>Nome</u> <u>Cognome</u> <u>NASCITA</u> <u>CF</u>						
---	--	--	--	--	--	--

Domini <u>Stringa</u> <u>Stringa</u> DATA <u>CodiceFiscale</u>						
--	--	--	--	--	--	--

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):

La relazione accorda le relazioni che implementano le seguenti associazioni:

[5] Relazione ..MEDICO..... (nome)	Derivante da: classe associazione (cerchiare)
---	---

Attributi <u>PERSONA</u> <u>PRIMARIA</u>						
--	--	--	--	--	--	--

Domini <u>CodiceFiscale</u> <u>Stringa</u>						
--	--	--	--	--	--	--

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(CF)

FK PRIMARIA REF SPECIALIZZAZIONE(none)

La relazione accorda le relazioni che implementano le seguenti associazioni:

6 Relazione PRESTAZIONE... (nome)				Derivante da: classe associazione (cerchiare)				
Attributi	<u>ID</u>	INIZIO	PAZIENTE	MEDICO				
Domini	Serial	DATAORA	CF	CF				

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(PERSONA) CHECK(PAZIENTE <> MEDICO)
FK MEDICO REF MEDICO(PERSONA)

La relazione accorda le relazioni che implementano le seguenti associazioni: PAZ_PRE, MED_PRE.....

7 Relazione .RICOVERO.... (nome)				Derivante da: classe associazione (cerchiare)				
Attributi	<u>PRESTAZIONE</u>	FINE *						
Domini	INT	DATAORA						

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 PRESTAZIONE REF PRESTAZIONE(ID)

CHECK(FINE <= NOW())

La relazione accorda le relazioni che implementano le seguenti associazioni: PRE_RIC.....

8 Relazione ESTERNO.... (nome)				Derivante da: classe associazione (cerchiare)				
Attributi	<u>PRESTAZIONE</u>	DESCRIZIONI	SPECIALIZZAZIONE					
Domini	INT	STRINGA	STRINGA					

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 PRESTAZIONE REF PRESTAZIONE(ID)

FK SPECIALIZZAZIONE REF SPECIALIZZAZIONE(NOME)

La relazione accorda le relazioni che implementano le seguenti associazioni: EST_PRE, EST_SPE.....

9 Relazione SPECIALIZZAZIONE (nome)				Derivante da: classe associazione (cerchiare)				
Attributi	<u>NOME</u>							
Domini	STRINGA							

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):

La relazione accorda le relazioni che implementano le seguenti associazioni:

10 Relazione MED-SPE.... (nome)				Derivante da: classe associazione (cerchiare)				
Attributi	<u>MEDICO</u>	SPECIALIZZAZIONE						
Domini	CF	STRINGA						

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 medico REF MEDICO(PERSONA)

FK SPECIALIZZAZIONE REF SPECIALIZZAZIONE(NOME)

La relazione accorda le relazioni che implementano le seguenti associazioni:

11 Relazione .LETTORI..... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>ID</u> <u>PIANO</u> <u>SETTORE</u>	
Domini 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):

CHECK (ID>0 AND ID<9)

FK (PIANO,settore) REF STANZA (PIANO,settore)

La relazione accorda le relazioni che implementano le seguenti associazioni: LET-STA.....

12 Relazione STANZA..... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>PIANO</u> <u>SETTORE</u>	
Domini INTPOS INTPOS	

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):

La relazione accorda 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 ennupla, di dominio):

La relazione accorda 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 ennupla, di dominio):

La relazione accorda 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 ennupla, di dominio):

La relazione accorda 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 ennupla, di dominio):

La relazione accorda 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 ennupla, di dominio):

La relazione accorda 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 ennupla, di dominio):

La relazione accorda 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 ennupla, di dominio):

La relazione accorda 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 ennupla, di dominio):

La relazione accorda 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, ennupla, dominio, inclusione), progettare un trigger che lo implementi, definendo: (a) gli eventi da intercettare (inserimento, modifica, eliminazione di ennuple); (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.

Trigger: Persona è completa

BEFORE INSERT OR UPDATE ON PERSONA

```
OK = Exist (Select 1 FROM PAZIENTE WHERE PERSONA = NEW.CF
UNION
Select 1 FROM MEDICO WHERE PERSONA = NEW.CF)
```

IF OK = TRUE

ELSE Permit

Raise error

V: Prestazione è completa

BEFORE INSERT OR UPDATE ON PRESTAZIONE

IF NOT EXIST (

```
Select 1 FROM ESTERNO WHERE PRESTAZIONE = NEW.ID
UNION
Select 1 FROM RICOVERO WHERE PRESTAZIONE = NEW.ID
```

) THEN

Raise error

IF EXISTS (

```
Select 1 FROM RICOVERO r Join ESTERNO e ON r.PRESTAZIONE = e.PRESTAZIONE WHERE r.PRESTAZIONE = NEW.ID
) THEN
```

Raise error

Permit

Trigger: Al massimo 8 letti per stanza

BEFORE INSERT ON LETTO

N = Select COUNT(*) FROM LETTO

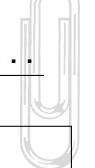
WHERE PIANO = NEW.PIANO AND SETTORE = NEW.SETTORE

IF N > 8 THEN

Raise error

Permit

Continua



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

CalcolaMediciCompetenti(S: Stringa): CodiceFiscale [0..*]

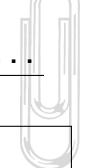
```
WITH primaria AS (
    SELECT m.PERSONA FROM medico m
    WHERE m.primaria = S
),
secondaria AS (
    SELECT m.PERSONA FROM medico m JOIN MED-SPEC-SPM ON m.PERSONA = S.m.medico
    WHERE S = ms.specializzazione
),
RESULT AS (
    SELECT * FROM primaria
    UNION ALL
    SELECT * FROM secondaria
    WHERE NOT EXISTS (SELECT 1 FROM primaria)
)
Select * FROM RESULT;
```

PostiDisponibili(): int

```
Select Count(*) FROM Letto l
    WHERE NOT EXISTS (SELECT 1 FROM Ricovero r
        WHERE r.letto = l.id AND r.fine IS NULL)
```

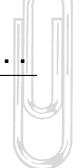
MostraItinerario(m: CodiceFiscale): (intpos, intpos)

```
SELECT DISTINCT(s.PIANO, s.settore) FROM STANZA s JOIN LETTO l ON l.PIANO = s.PIANO AND l.settore = s.settore
    JOIN RICOVERO R ON l.id = r.letto AND r.FINE IS NULL
    JOIN PRESTAZIONE p ON p.id = r.prestazione AND p.medico = m
    ORDER BY s.PIANO, s.settore ASC
```



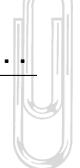
Risposta alla Domanda 8 (segue)

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.]

[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.]