

2

Specifiche 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, del pazienti interessano alcune informazioni anagrafiche (nome, cognome e data di nascita) ed i loro 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.

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 letto occupati durante i diversi ricoveri.¹

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 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 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 $[(1,1), (1,3), (3,4), (7,1), (7,7)]$.

¹Si assume 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.

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

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.

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



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.)

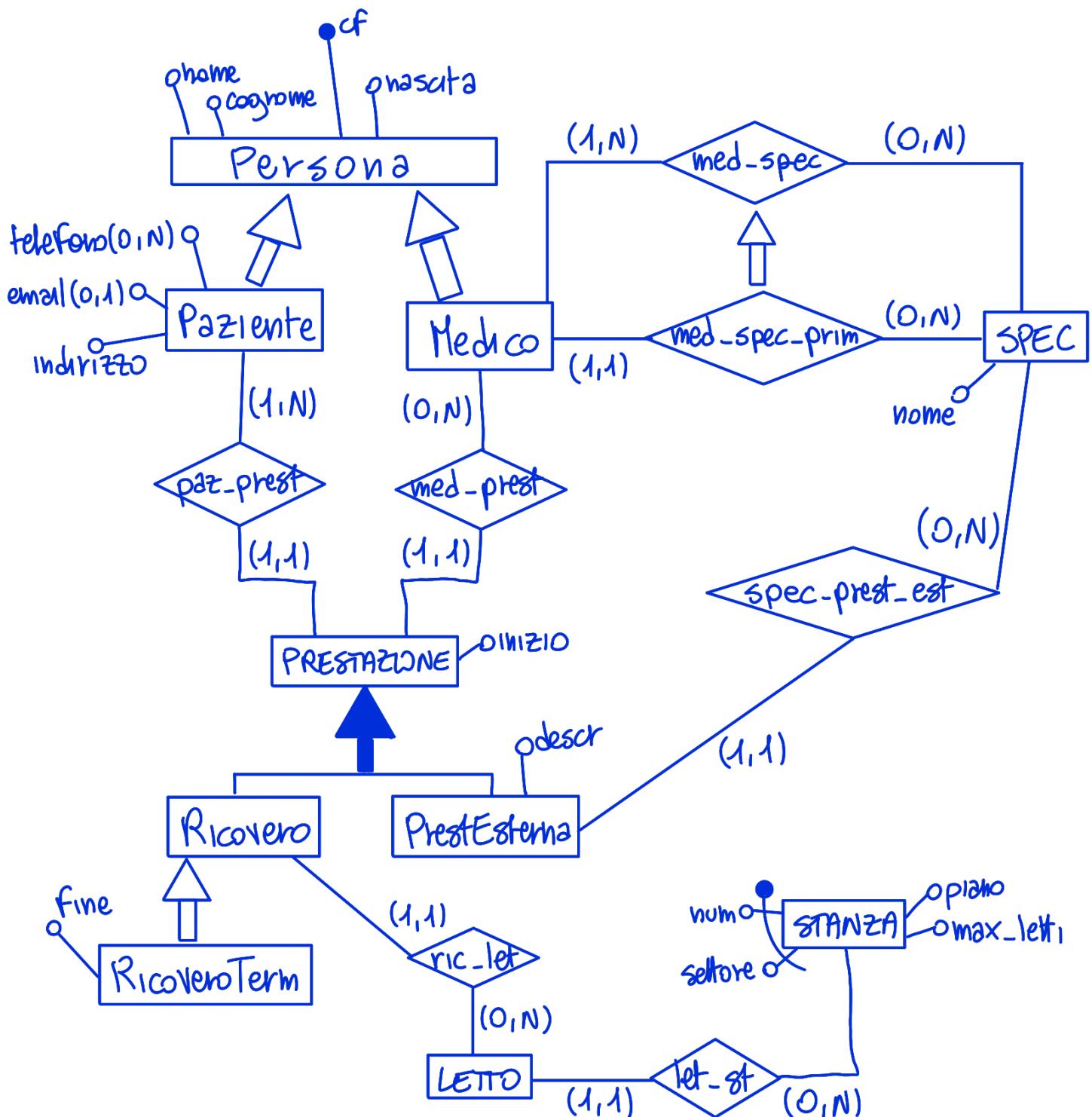
Risposta

Domanda 2 (45 minuti; 75 minuti al massimo) Proseguire la fase di Analisi Concettuale dei requisiti, producendo un diagramma ER concettuale per l'applicazione, il dizionario dei dati ed eventuali vincoli esterni.

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

Diagramma ER

Produrre un diagramma ER concettuale per l'applicazione in termini di entità, relationship, attributi, relazioni is-a, generalizzazioni (disgiunte) complete e non.



Dizionario dei dati Per ogni entità e relationship del diagramma ER **con** attributi o vincoli:

- Definire il dominio e la molteplicità degli attributi (se diversa da (1,1))
- Definire eventuali vincoli esterni in logica del primo ordine estesa con teoria degli insiemi e semantica di mondo reale, usando il seguente alfabeto:
 - Un simbolo di predicato $E/1$ per ogni entità E .
Semantica di $E(x)$: x è una istanza di E .
 - Un simbolo di predicato $D/1$ per ogni dominio D .
Semantica di $D(x)$: x è un valore di D .
 - Un simbolo di predicato r/n ($n > 0$) per ogni relationship n -aria r .
Semantica di $r(x_1, \dots, x_n)$: x_1, \dots, x_n è una istanza di r .
 - Un simbolo di predicato $a/2$ per ogni attributo a di entità
Semantica di $a(x, v)$: uno dei valori dell'attributo a dell'istanza x è v .
 - Un simbolo di predicato $a/(n+1)$ per ogni attributo a di relationship n -aria.
Semantica di $a(x_1, \dots, x_n, v)$: uno dei valori dell'attr. a dell'istanza (x_1, \dots, x_n) della relat. è v .
 - Opportuni simboli di predicato (soggetti a *semantica di mondo reale*) per gestire confronti tra valori di domini numerici o comunque ordinati (tra cui $</2$, $\leq/2$, $>/2$, $\geq/2$).
 - Il predicato di uguaglianza $=/2$ (la cui interpretazione è la relazione che lega ogni elemento del dominio di interpretazione solo con se stesso).
 - Opportuni simboli di costante (soggetti a *semantica di mondo reale*), tra cui *adesso*, interpretato come il valore del dominio DataOra che rappresenta l'istante corrente.

Risposta

<p>1 Tipo: Entità Relationship (cerchiare) Nome: RicoveroTerm</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">attributo</th><th style="text-align: left;">dominio</th><th style="text-align: left;">moltep. (*)</th></tr> </thead> <tbody> <tr> <td style="text-align: left;">fine</td><td style="text-align: left;">data</td><td></td></tr> </tbody> </table> <p>(*) solo se diversa da (1,1) Vincoli: $[(\forall r, f, i) \ RicoveroTerm(r) \wedge fine(f, r) \wedge inizio(i, r) \rightarrow f > i]$ </p>	attributo	dominio	moltep. (*)	fine	data		<p>2 Tipo: Entità Relationship (cerchiare) Nome: PrestEsterna</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">attributo</th><th style="text-align: left;">dominio</th><th style="text-align: left;">moltep. (*)</th></tr> </thead> <tbody> <tr> <td style="text-align: left;">desc</td><td style="text-align: left;">stringa</td><td></td></tr> </tbody> </table> <p>(*) solo se diversa da (1,1) Vincoli:</p>	attributo	dominio	moltep. (*)	desc	stringa	
attributo	dominio	moltep. (*)											
fine	data												
attributo	dominio	moltep. (*)											
desc	stringa												

15	Tipo: Entità Relationship (cerchiare)	
Nome:	Persona	
attributo	dominio	moltep. (*)
cf	CodFisc	
nome	stringa	
cognome	stringa	
nascita	data	

(*) solo se diversa da (1,1)

Vincoli:

[V. Persona.disj]

$$\forall p \text{ Persona}(p) \rightarrow \\ \text{Paziente}(p) \vee \text{Medico}(p)$$

17	Tipo: Entità Relationship (cerchiare)	
Nome:	Paziente	
attributo	dominio	moltep. (*)
indirizzo	Indirizzo	
email	Email	(0,1)
telefono	Telefono	(0,N)

(*) solo se diversa da (1,1)

Vincoli:

16	Tipo: Entità Relationship (cerchiare)	
Nome:	PRESTAZIONE	
attributo	dominio	moltep. (*)
inizio	date	

(*) solo se diversa da (1,1)

Vincoli:

[V. Prestaz.med - no - paz]

$$\forall prest, p \text{ Prestazione}(prest) \wedge \\ \text{med-prest}(p, prest) \\ \rightarrow \text{paz-prest}(p, prest)$$

18	Tipo: Entità Relationship (cerchiare)	
Nome:	SPECIALIZZAZIONE	
attributo	dominio	moltep. (*)
nome	stringa	

(*) solo se diversa da (1,1)

Vincoli:

<p><input type="checkbox"/> 3 Tipo: Entità Relationship (cerchiare)</p> <p>Nome: <u>stanza</u></p> <table border="1"> <thead> <tr> <th>attributo</th><th>dominio</th><th>moltep. (*)</th></tr> </thead> <tbody> <tr> <td>num</td><td>intero > 0</td><td></td></tr> <tr> <td>settore</td><td>intero > 0</td><td></td></tr> <tr> <td>piano</td><td>intero ≥ 0</td><td></td></tr> <tr> <td>max-letti</td><td>int [1, 8]</td><td></td></tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p> <p>$\exists s, m$ $stanza(s) \wedge max-letti(s, m)$ $\rightarrow \{x \mid let-st(x, s)\} \leq m$</p>	attributo	dominio	moltep. (*)	num	intero > 0		settore	intero > 0		piano	intero ≥ 0		max-letti	int [1, 8]		<p><input type="checkbox"/> 5 Tipo: Entità Relationship (cerchiare)</p> <p>Nome: <u>letto</u></p> <table border="1"> <thead> <tr> <th>attributo</th><th>dominio</th><th>moltep. (*)</th></tr> </thead> <tbody> <tr> <td></td><td></td><td></td></tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p>	attributo	dominio	moltep. (*)			
attributo	dominio	moltep. (*)																				
num	intero > 0																					
settore	intero > 0																					
piano	intero ≥ 0																					
max-letti	int [1, 8]																					
attributo	dominio	moltep. (*)																				

<p><input type="checkbox"/> 4 Tipo: Entità Relationship (cerchiare)</p> <p>Nome:</p> <table border="1"> <thead> <tr> <th>attributo</th><th>dominio</th><th>moltep. (*)</th></tr> </thead> <tbody> <tr> <td></td><td></td><td></td></tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p>	attributo	dominio	moltep. (*)				<p><input type="checkbox"/> 6 Tipo: Entità Relationship (cerchiare)</p> <p>Nome:</p> <table border="1"> <thead> <tr> <th>attributo</th><th>dominio</th><th>moltep. (*)</th></tr> </thead> <tbody> <tr> <td></td><td></td><td></td></tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p>	attributo	dominio	moltep. (*)			
attributo	dominio	moltep. (*)											
attributo	dominio	moltep. (*)											

Ulteriori vincoli esterni, specifica di eventuali operazioni ausiliarie invocate da tali vincoli, e specifica dei domini concettuali non di tipo base

[V. Prest. med-e-paz-nati]

$\forall \text{pers}, \text{prest}, n, i$

$\text{Persona}(p) \wedge \text{Prestazione}(\text{prest})$

$\wedge [\text{paz-prest}(p, \text{prest}) \vee \text{med-prest}(p, \text{prest})] \wedge$
 $\text{nascita}(n, p) \wedge \text{inizio}(\text{prest}, i) \rightarrow n < i$

[V. letto.nonCondiviso]

$\text{Ricovero}(r) \wedge \text{Ricovero}(r') \wedge \text{Letto}(l) \wedge r \neq r' \wedge \text{ricLet}(r, l)$
 $\wedge \text{ricLet}(r', l) \wedge \text{inizio}(r, i) \wedge \text{inizio}(r', i') \rightarrow$

$\neg \left[\begin{array}{l} \exists \text{ Data}(t) \wedge \\ t \geq i \wedge \\ (\forall f \text{ fine}(r, f) \rightarrow t \leq f) \wedge \\ t \geq i' \wedge \\ (\forall f' \text{ fine}(r', f') \rightarrow t \leq f') \end{array} \right]$

Domino Email

- secondo stringa standard

Risposta alla Domanda 2 (segue)

Dominio CF

- stringa di 16 caratteri secondo standard

Dominio Telefono

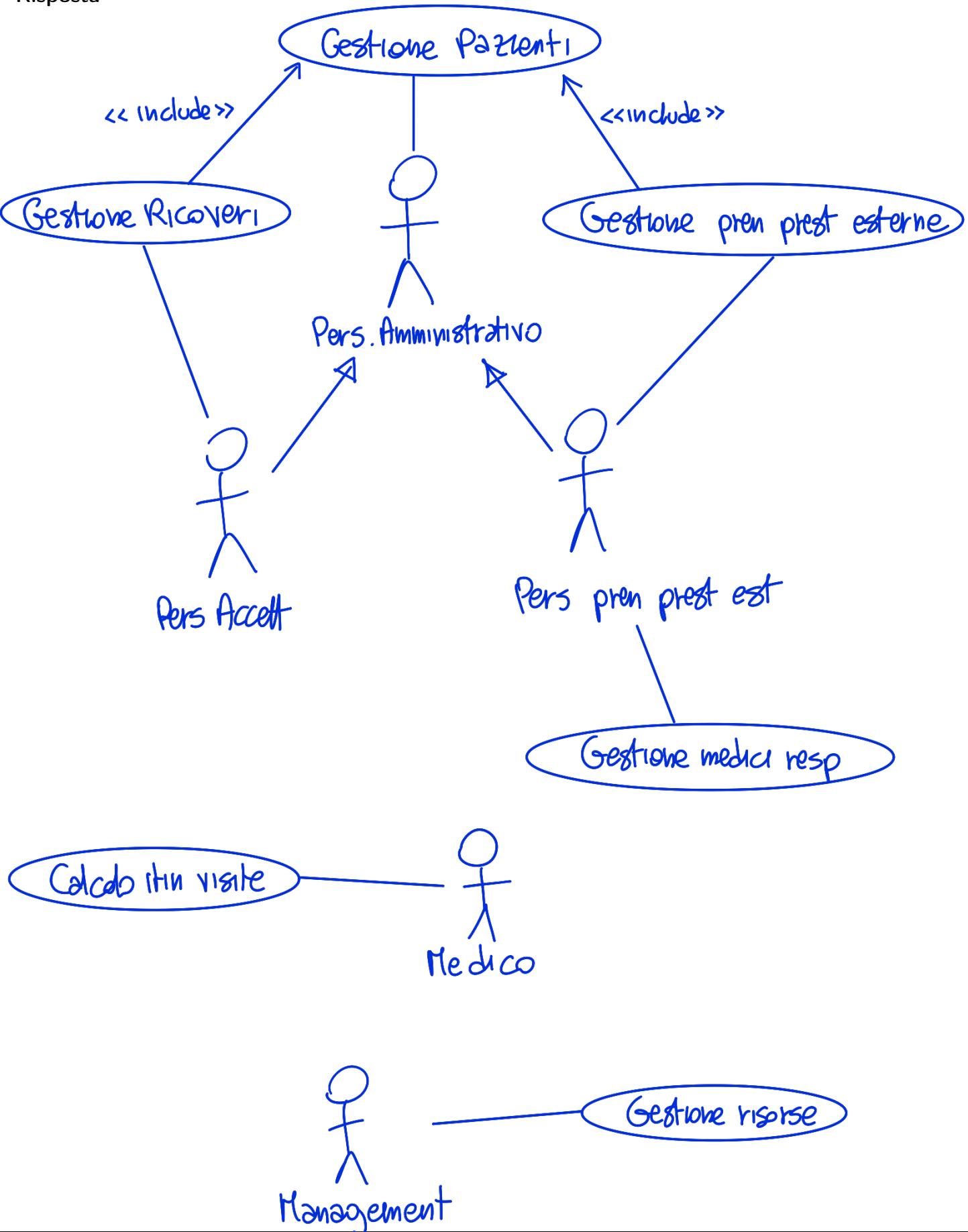
- stringa secondo standard

Dominio Indirizzo

- via : str
- civico : int > 0 (0,1)
- cap : stringa di 5 cifre
- città : str
- nazione : str

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



Domanda 4 (10 minuti) Proseguire la fase di Analisi Concettuale dei requisiti definendo le operazioni degli use-case.

In particolare, per ogni use-case definito nella risposta alla **Domanda 3** definire la **segnatura** di tutte le operazioni che lo compongono, in termini di nome dell'operazione, nomi e dominio concettuale degli argomenti, dominio concettuale dell'eventuale valore di ritorno.

1 Specifica use-case: Gestione pazienti (nome use-case)

Operazioni dello use-case:

registraPaziente (cf:CodFisc, na:data, n0:str, co:str, i:Indirizzo
t(0,N):Telefono, em(0,1):Email) : Paziente

storicoPazienti (p:Paziente) : Prestazione(0,N)

2 Specifica use-case: Gestione ricoveri (nome use-case)

Operazioni dello use-case:

ammettiPaziente (p:Paziente, d:dataora, m:Medico) : Ricovero

dimettiPaziente (r:Ricovero)

numPostiDisponibili (s:Stanza, d:dataora) : int ≥ 0

3 Specifica use-case: Gestione pren. prest. esterne (nome use-case)

Operazioni dello use-case:

registraPrestEsterna (p:Paziente, s:Specializzazione, d:stringa,
d':dataora) : PrestEsterna

4 Specifica use-case: Calcolo_itin_visite (nome use-case)

Operazioni dello use-case:

calcoloPercorso (m:Medico): (s:Stanza, n:Intero > 0) (O,N)

sortByPianoSettore(s:Stanza, s':Stanza): Boolean

5 Specifica use-case: Gestione medici responsabili (nome use-case)

Operazioni dello use-case:

medici_idolhei (s:Specializzazione): Medici (O,N)

6 Specifica use-case: Gestione risorse (nome use-case)

Operazioni dello use-case:

registraMedico(no:str, co:str, na:data, cf:CF, s:Specializzazione): Medico

aggiungiLetto(s:Stanza): Letto

rimuoviLetto(l:Letto)

7 Specifica use-case: (nome use-case)

Operazioni dello use-case:

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). 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

Specifico UseCase: GestioneRicoveri

numeroPostiDisponibili(s: Stanza, d: data): intero ≥ 0

precondizioni: nessuna

postcondizioni:

Modifica del liv. estens. dei dati: nessuna

Valore di ritorno: Sia

$$L = \{(l) \mid \text{let-st}(l, s) \wedge (\exists r, i \text{ Ricovero}(r) \wedge \text{ric-letto}(r, l) \wedge \text{RicoveroTerm}(r) \wedge \text{inizio}(i, r) \wedge i \leq d)\}$$

$$\text{result} = |L|$$

ammettiPaziente(p: Paziente, d: dataora, m: Medico): Ricovero

precondizioni: $\exists s \text{ Stanza}(s) \wedge \underline{\text{numPostiDisponibili}(s, d) > 0}$

postcondizioni: Sia l t.c: letto(l) $\wedge (\exists s \text{ Stanza}(s) \wedge \text{stLetto}(s, l) \wedge \overbrace{\text{Mod. del liv. estens. dei dati: si}}^{\text{l'arrivo del paziente}})$

Mod. del liv. estens. dei dati: si

Var. nel dom. di interpret: nuova tupla d

Var. tupla di pred.: Ricovero(d), ric-let(d, l),
paz-prest(p, d), med-prest(m, d),
inizio(d, d)

Val. ritorno: result=d

Risposta alla Domanda 5 (segue)

Specifico UseCase: Gestione medici responsabili

medici_idohel (s: Specializzazione): Medici(0, N)

precondizioni: nessuna

postcondizioni:

Modifica del liv. estens. dei dati: nessuna

Valore di ritorno: Siano:

$$M_p = \{ m \mid \text{med-spec-prim}(s, m) \}$$

$$M_s = \{ m \mid \text{med-spec}(s, m) \}$$

$$\text{result} = \begin{cases} M_p & \text{se } M_p \neq \emptyset \\ \text{altrimenti } M_s \end{cases}$$

Specifico UseCase: Calcolo itinerario visite

sortByPianoSettore(s: Stanza, s': Stanza): Boolean

precondizioni: nessuna

postcondizioni:

Modif. del liv. estensionale dei dati: nessuna

Valore di ritorno: Siano: p, p', st, st' t.c. vale:

$$\text{piano}(p, s) \wedge \text{piano}(p', s') \wedge \text{settore}(s, st) \wedge \text{settore}(s', st')$$

$$\left. \begin{array}{l} ((p < p' \vee (p = p' \wedge st < st')) \rightarrow \text{result} = \text{TRUE}) \\ \wedge \\ ((p > p' \vee (p = p' \wedge st > st')) \rightarrow \text{result} = \text{FALSE}) \end{array} \right\}$$

Risposta alla Domanda 5 (segue)

calcoloPercorso (m: Medico) : (s: Stanza, n: Intero > 0) (O, N)

precondizioni: nessuna

postcondizioni:

Mod. del liv. estens. dei dati: nessuna

Valore di ritorno: Sia

$$S = \{ s \mid \text{stanza}(s) \wedge (\exists l, r \text{ let-st}(l, s) \wedge \\ \text{ric-let}(l, r) \wedge \text{med-prest}(m, r) \\ \wedge \neg \text{RicoveroTerm}(r) \}$$

result = Sorted(S, sortByPianoSettore)

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 ER concettuale, il dizionario dei dati e i vincoli esterni. In particolare:

- progettare una corrispondenza tra i domini 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 entità
- valutare se e come aggiungere ridondanza in maniera controllata
- 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 domini concettuali e domini supportati dal DBMS

```

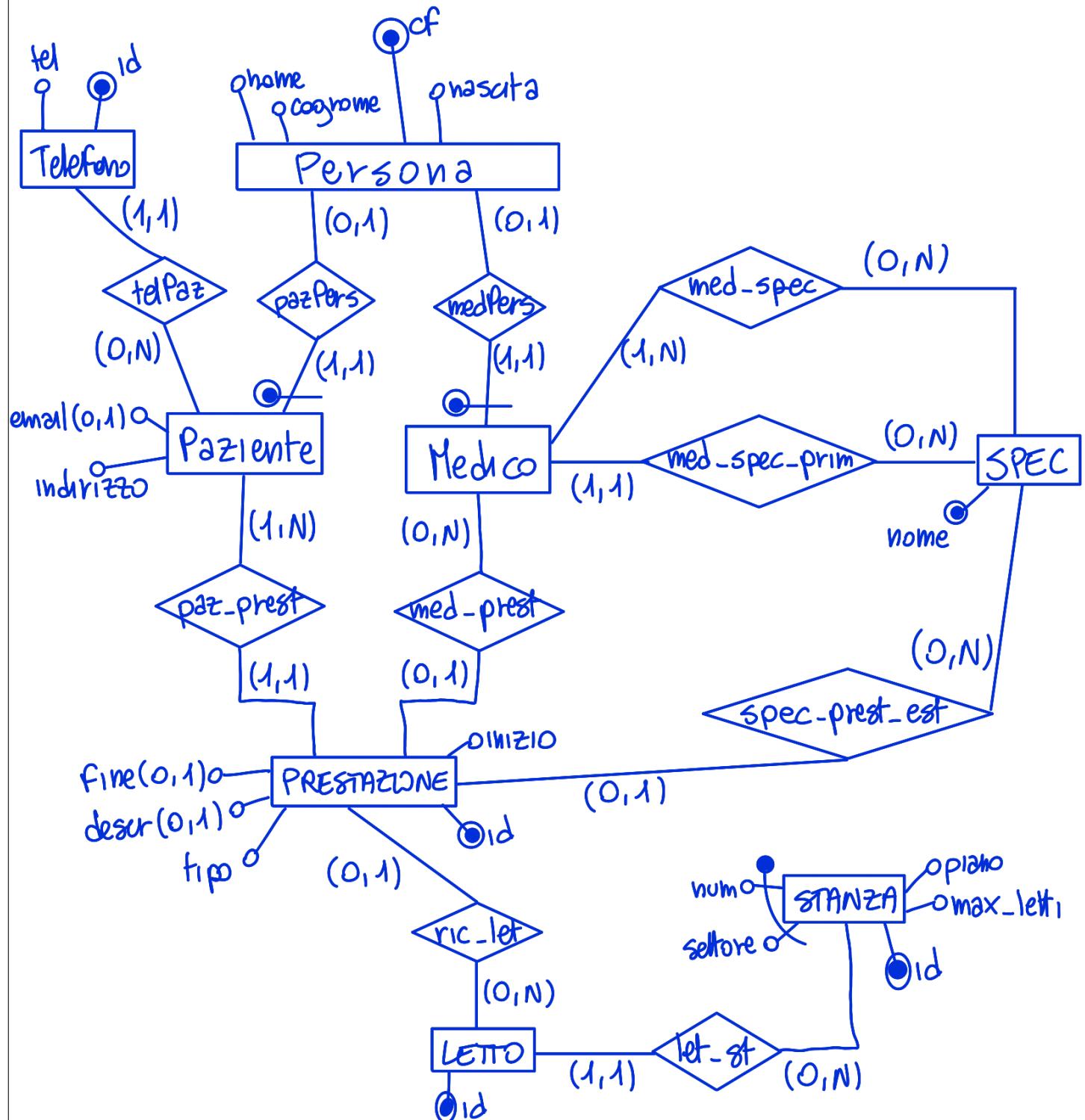
create domain StringS as varchar(50);
create domain StringM as varchar(200);
create domain StringL as varchar(1000);
create domain CodiceFiscale as char(16) check (isValidCF(value));
create domain Email as StringS check (isValidEmail(value));
create domain IntegerGEZ as integer check (value ≥ 0);
create domain IntegerGZ as integer check (value > 0);
create domain Max_Posti as integer check (value ≥ 1 and value ≤ 8);

create type Indirizzo as (
    Via : StringS
    CIVICO : IntegerGZ (0,1)
    CAP : IntegerGZ
    Città : StringS
    Nazione : StringS
);

create type Tipo_Prest as
enum ('Ricon', 'PrEster')

```

Diagramma ER ristrutturato



Breve descrizione delle scelte effettuate durante la ristrutturazione

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. med-spec-prima.15a]

$\forall m, s \quad \text{med-spec-prima}(m, s) \rightarrow \text{med-spec}(m, s)$

[V. Persona.complettezza] (Trigger)

$\forall x \quad \text{Persona}(x) \rightarrow$

$[\exists p \quad \text{paZPers}(x, p)] \rightarrow [\nexists m \quad \text{medPers}(m, x)]$

\wedge

$[\exists p \quad \text{paZPers}(x, p)] \vee [\exists m \quad \text{medPers}(m, x)]$

Risposta alla Domanda 6 (segue)

[V.Prestazione.Ricovero]

$\forall p \text{ Prestazione}(p) \rightarrow$

$[\exists l \text{ ric-let}(p,l) \leftrightarrow \text{tip}(p, 'Ricov')]$

$\wedge [\exists s,d \text{ spec-prest-est}(p,s) \wedge \text{desc}(p,d) \leftrightarrow \text{tip}(p, 'Prester')]$

[V.Prestazione.Fine]

$\forall p \text{ Prestazione}(p) \rightarrow [\exists f \text{ fine}(p,f) \leftrightarrow \text{tip}(p, 'Ricov')]$

[V.Ricovero.Term.Fine]

$\forall p,i,f \text{ Prestazione}(p) \wedge \text{inizio}(i,p) \wedge \text{fine}(f,p) \rightarrow i < f$

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 ER ristrutturato.

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

1	Relazione	<u>Telefono</u> (nome)	Derivante da:	entità	relationship (cerchiare)
Attributi	<u>id</u>	<u>tel</u>	<u>pa</u> ziente		
Domini	integer	Telefono	CodFisc		

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):

seriale: i valori di id sono generati automaticamente dal DBMS

FK: paziente refer Paziente(cf)

La relazione accorda le relazioni che implementano le seguenti relationship: ... *EdPaz*

2	Relazione	<u>Paziente</u> (nome)	Derivante da:	entità	relationship (cerchiare)
Attributi	<u>Indirizzo</u>	<u>email</u> *	<u>per</u> sona		
Domini	Indirizzo	Email	CodFisc		

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 refer Persona(cf)

La relazione accorda le relazioni che implementano le seguenti relationship: ... *pazPers*

3	Relazione	<u>Medico</u> (nome)	Derivante da:	entità	relationship (cerchiare)
Attributi	<u>per</u> sona	<u>specia</u> lizz.			
Domini	CodFisc	StringM			

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 refer Persona(cf)

FK: specializzazione refer Specializzazione(nome)

La relazione accorda le relazioni che implementano le seguenti relationship: ... *medPers, med-spec-prim*

4	Relazione	<u>Person</u> a (nome)	Derivante da:	entità	relationship (cerchiare)
Attributi	<u>cf</u>	<u>nome</u>	<u>cognome</u>	<u>nascita</u>	
Domini	CodFisc	StringM	StringM	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):

La relazione accorda le relazioni che implementano le seguenti relationship:

5	Relazione	<u>Prestazione</u> ... (nome)	Derivante da:	entità	relationship (cerchiare)
Attributi	<u>id</u>	<u>tipo</u>	<u>fine</u>	<u>inizio</u> *	<u>desc</u> *
Domini	integer	Tipo-PRES	date	date	StringL

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): *(specializzazione: StringM)* ——————
FK: paziente ref Paziente(cf) *PK: medico ref Medico(cf)* *seriale: id gener autom. del DBMS*

FK: letto ref Letto(id) *PK: spec refer Spec(nome)* *(Vincoli A PAG MINUTE 1)*

La relazione accorda le relazioni che implementano le seguenti relationship: ... *paz.Prest, med_Prest, ric-let, sp-prest, est*

6 Relazione	<u>Lett</u> (nome)	Derivante da:	entità	relationship (cerchiare)
Attributi	<u>id</u>	<u>stanza</u>		
Domini	<u>integer</u>	<u>integer</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): *senza: id gener autom. dal DBMS*
FK: stanza refer stanza(id)

La relazione accorda le relazioni che implementano le seguenti relationship: *st-lst*

7 Relazione	<u>stanza</u> (nome)	Derivante da:	entità	relationship (cerchiare)
Attributi	<u>id</u>	<u>settore</u>	<u>num</u>	<u>piano</u>
Domini	<u>integer</u>	<u>IntegerGZ</u>	<u>IntegerGZ</u>	<u>IntegerGZ</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): *senza: id gener autom. dal DBMS*
Inclusione: id ⊆ letto(stanza)

La relazione accorda le relazioni che implementano le seguenti relationship:

8 Relazione	<u>spe-med</u> (nome)	Derivante da:	entità	relationship (cerchiare)
Attributi	<u>medico</u>	<u>specializz.</u>		
Domini	<u>CodFisc</u>	<u>String&M</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: specializzazione refer Specializzazione(nome)

FK: medico refer Persona(cf)

La relazione accorda le relazioni che implementano le seguenti relationship:

9 Relazione (nome)	Derivante da:	entità	relationship (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 relationship:

10 Relazione (nome)	Derivante da:	entità	relationship (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 relationship:

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.

[V.Stanza.maxLetti]

Operazione: inserimento in Letto

Istante: prima dell'operazione intercettata

Funzione:

- 1 $Q = (\text{select count}(l.id) \text{ as } c, s.\text{max_letti} \text{ as } m$
 $\text{from Letti } l, Stanza } s$
 $\text{where } l.\text{stanza} = \text{new}.stanza$
 $\text{and } l.\text{stanza} = s.\text{id}$
 $\text{group by } s.\text{max_letti}$

- 2 if $Q.c < Q.m$: commit

- 3 else: genera errore

[V.Persone.completezza] (Trigger)

$$\forall x \text{ Persona}(x) \rightarrow$$

$$[\exists p \text{ pazfers}(x, p)] \rightarrow [\exists m \text{ medPers}(m, x)]$$

$$\wedge$$

$$[\exists p \text{ pazfers}(x, p)] \vee [\exists m \text{ medPers}(m, x)]$$
[V.Persone.Completezza]

Operazioni: inserimento

Istante: prima dell'operazione intercettata

Funzione:

- 1 $isValid = (\text{exist (select * from Paziente P, Medico m}$

$$\text{where p.persona} = \text{new.cf} \text{ and m.pers} = \text{new.cf})$$

- 2 if $isValid$: commit

- 3 else: genera errore

Domanda 8 (30 minuti; 45 minuti al massimo) Proseguire la fase di progettazione dell'applicazione producendo le specifiche realizzative delle operazioni di 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.

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

Risposta

`calcolaMediciIdonei(S: String M) : Insieme(<m: CodiceFiscale>)`

`Q = (select m.id
 from Medici m
 where m.specializzazione =: S)`

`if Q ≠ NULL : return Q`

`else:`

`Q'=(select sp.Medico as M
 from spe-med sp
 where sp.specializzazione =: S)`

`return Q'`

Risposta alla Domanda 8 (segue)

calcolaPercorso(m: CodiceFiscale): Lista(<s:integer>)

Q = (select s.id
from Stanza s, letto l, Postazione p
where p.tipo='Ricov'
and p.letto=l.id
and p.medico=:m
order by (s.piano, s.settore))

return Q

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.]

ennupla: inizio < fine

ennupla: desc ≠ NULL \leftrightarrow TIPO = 'PrEster'

ennupla: specializzazione ≠ NULL \leftrightarrow TIPO = 'PrEster'

ennupla: paziente ≠ medico

ennupla: tipo 'RIC' \leftrightarrow LETTO ≠ NULL

ennupla: fine ≠ NULL \rightarrow tipo = 'RIC'