



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.ER** – Modulo risposte prova scritta

Dati dello studente e dell'esame

Cognome e nome: Matricola:

Data:

Corso di laurea e canale di appartenenza:

- ☐ Laurea in Informatica, canale 1 (A-L, Prof. G. Perelli)
- ☐ Laurea in Informatica, canale 2 (M-Z, Prof.ssa M. De Marsico)
- ☐ Laurea in Informatica in Modalità Teledidattica Unitelma Sapienza

Firma di un membro della Commissione per
avvenuta identificazione:

.....

Rinuncia alla prova

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



Questo modulo è ottimizzato per la stampa fronte-retro



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 ER concettuale (*)
 - il relativo dizionario dei dati
 - le specifiche dei domini concettuali non di tipo base
 - eventuali vincoli esterni, espressi utilizzando il linguaggio della logica del primo ordine (*)
- 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)
 - la specifica delle operazioni di use-case necessarie a modellare i requisiti contrassegnati dalla barra laterale (come quella qui a sinistra) in termini di precondizioni e postcondizioni, utilizzando il linguaggio della logica del primo ordine (*) (30 minuti; 60 minuti al massimo)

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 ER concettuale e dei vincoli esterni (20 minuti; 30 minuti al massimo):
 - * scelta del DBMS da utilizzare
 - * progettazione della corrispondenza tra i domini concettuali ed opportuni domini SQL (domini base o utente, oppure realizzati mediante relazioni aggiuntive) supportati dal DBMS scelto
 - * ristrutturazione del diagramma ER concettuale e 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 di use-case 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. (*)

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

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 33 in poi possono essere utilizzate per scrivere minute che non verranno valutate.



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

1. Utente

- 1.1 nome
- 1.2 cognome
- 1.3 email
- 1.4 città di provenienza
- 1.5 data iscrizione

2. Viaggio

- 2.1 nome
- 2.2 creatore
- 2.3 min partecipanti
- 2.4 max partecipanti
- 2.5 partecipanti
- 2.6 attività
- 2.7 budget()
 - somma prezzi delle attività
- 2.8 inizio()
- 2.9 fine()

→ e Feedback (1..5)

3. Attività

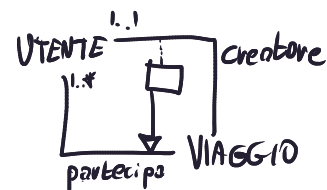
- 3.1 nome
- 3.2 istante inizio
- 3.3 durata in minuti
- 3.4 prezzo
- 3.5 info testuali
- 3.6 utente che vi partecipa (eventuale biglietto)
- 3.7 Luogo
 - 3.7.1 indirizzo
 - 3.7.2 città
 - 3.7.3 regione
 - 3.7.4 nazione
- 3.8 istante_fine(): inizio + durata
- 3.9 tipo speciale? → {dis}

4. Attività composte

- 4.1 insieme di attività
- 4.2 prezzo_tot()

3.9.2 Spostamento

- 3.9.2.1 luogo arrivo
- 3.9.2.2 mezzo
- 3.9.3 Prenotamento



V. un utente non
partecipa contemporaneamente
a due viaggi

V. utente nasce
prima di ogni attività/viaggi
alla quale partecipa

V. no attività che si
intersecano in uno
stesso
viaggio

(controlla composte)

V. utente partecipa ad
una attività compresa
nel viaggio alla quale
partecipa.

V. se un attività richiede un
biglietto, e' e n'e' uno per
utente coinvolto, altrimenti
0.

V. creatore non può modificare
uno stesso viaggio

V. tutte le attività
di un viaggio
si svolgono fra
l'inizio e la fine
del viaggio

V. viene dato un
numero parteciperà

$V. \max Part \geq \min Part$

V. no due prenotazioni
nella stessa data

V. le attività composte di un
viaggio hanno attività di quel
viaggio

V. le attività si svolgono in
tempi compresi nel viaggio

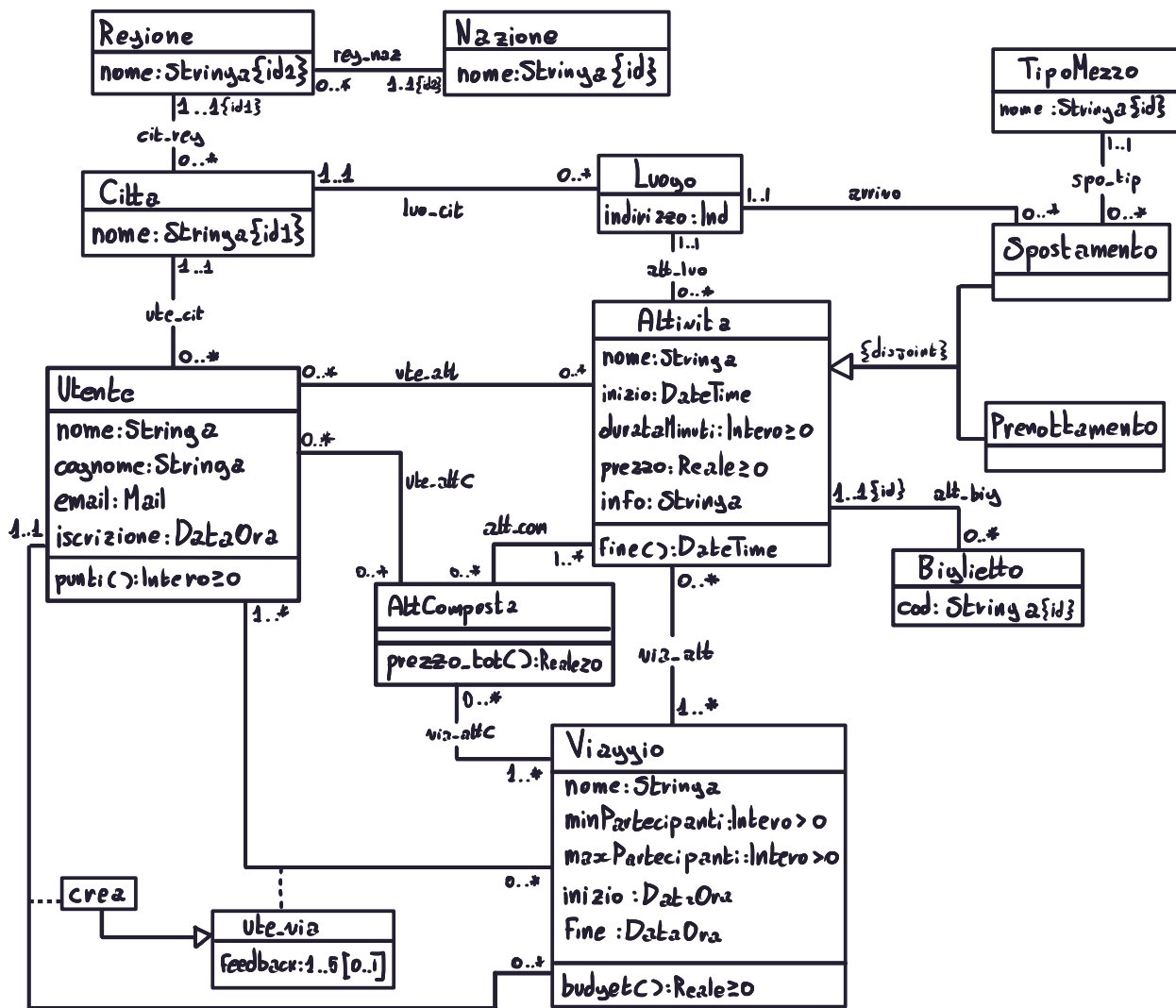
Risposta alla Domanda 1 (segue)

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

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



Risposta alla Domanda 2 (segue)

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

<div>1</div> <div>Tipo: <u>Entità</u> Relationship (cerchiare)</div> <div>Nome: <u>Utente</u></div> <div><table><tr><td>attributo</td><td>dominio</td><td>moltepl. (*)</td></tr><tr><td colspan="3"></td></tr></table></div> <div>(*) solo se diversa da (1,1)</div> <div>Vincoli:</div> <div><u>[V.no-2-viaggi]</u></div> <div>$\forall u, v_1, v_2, i_1, i_2, f_1, f_2$$[Utente(u) \wedge Viaggio(v_1) \wedge Viaggio(v_2) \wedge ute_via(u, v_1) \wedge u \neq v_2$$ute_via(u, v_2) \wedge inizio(v_1, i_1) \wedge inizio(v_2, i_2) \wedge Fine(v_1, f_1)$$\wedge Fine(v_2, f_2)] \rightarrow [f_1 < i_2 \vee f_2 < i_2]$</div> <div><u>[V.iscritto-per-viaggi]</u></div> <div>$\forall u, v, i, n$$[Utente(u) \wedge Viaggio(v) \wedge iscrizione(u, n) \wedge inizio(v, i)$$\wedge ute_via(u, v)] \rightarrow n < i$</div>	attributo	dominio	moltepl. (*)				<div>2</div> <div>Tipo: <u>Entità</u> Relationship (cerchiare)</div> <div>Nome: <u>Utente</u></div> <div><table><tr><td>attributo</td><td>dominio</td><td>moltepl. (*)</td></tr><tr><td colspan="3"></td></tr></table></div> <div>(*) solo se diversa da (1,1)</div> <div>Vincoli:</div> <div><u>[V.no-auto-valutazioni]</u></div> <div>$\forall u, v$$[Utente(u) \wedge crea(u, v)] \rightarrow [\neg \exists f feedback(u, v, f)]$</div> <div><u>[V.partecipa-att-del-viaggio]</u></div> <div>$\forall u, a [Utente(u) \wedge ute_att(z, u)] \rightarrow [\exists v Viaggio(v)$$ute_via(u, v) \wedge [via_att(v, a) \vee [\exists ac via_att(v, ac)$$\wedge att_com(z, ac)]]]$</div>	attributo	dominio	moltepl. (*)			
attributo	dominio	moltepl. (*)											
attributo	dominio	moltepl. (*)											

3 Tipo: Entità | Relationship (cerchiare)Nome: Viaggio

attributo	dominio	moltepl. (*)

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

Vincoli:

[V.inizio-poi-fine] $\forall v, i, f [Viaggio(v) \wedge inizio(v, i) \wedge fine(v, f)] \rightarrow i < f$ [V.max-maggiore-min] $\forall v, max, min [Viaggio(v) \wedge maxPartecipanti(v, max) \wedge minPartecipanti(v, min)] \rightarrow max \geq min$ [V.limite-partecipanti] $\forall v, m [Viaggio(v) \wedge maxPartecipanti(v, m)] \rightarrow |\{u | ute.via(u, v)\}| \leq m$ 5 Tipo: Entità | Relationship (cerchiare)Nome: Prenotamento

attributo	dominio	moltepl. (*)

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

Vincoli:

[V.no-stessa-data] $\forall u, p_1, p_2, i_1, i_2, d_1, d_2$ $[p_1 \neq p_2 \wedge Utente(u) \wedge Prenotamento(p_1) \wedge Prenotamento(p_2) \wedge ute_att(u, p_1) \wedge ute_att(u, p_2) \wedge inizio(p_1, i_1) \wedge inizio(p_2, i_2) \wedge Data(i_1, d_1) \wedge Data(i_2, d_2)] \rightarrow d_1 \neq d_2$ 4 Tipo: Entità | Relationship (cerchiare)Nome: Viaggio

attributo	dominio	moltepl. (*)

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

Vincoli:

[V.attività-dentro-tempi-viaggio] $\forall v, iv, fv, z, i_2, f_2$ $[Viaggio(v) \wedge inizio(v, iv) \wedge fine(v, fv) \wedge Attività(z) \wedge inizio(z, i_2) \wedge fine(z, f_2) \wedge [via_att(v, z) \vee [\exists ac via_att(v, ac) \wedge att_com(z, ac)]]] \rightarrow [iv \leq i_2 \wedge fv \geq f_2]$ 6 Tipo: Entità | Relationship (cerchiare)Nome: AttComposta

attributo	dominio	moltepl. (*)

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

Vincoli:

[V.inclusione-in-viaggio] $\forall z, ac, v$ $[Viaggio(v) \wedge AttComposta(z) \wedge via_att(v, z) \wedge att_com(z, ac)] \rightarrow via_att(v, ac)$

7 Tipo: ~~Entità~~ | Relationship (cerchiare)Nome: Viaggio

attributo	dominio	moltepl. (*)
-----------	---------	--------------

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

Vincoli:

[V.no_intersezione_attivita] $\forall u, z1, z2, i1, i2, f1, f2$ $[Viaggio(u) \wedge Attivita(z1) \wedge Attivita(z2) \wedge z1 \neq z2$ $\wedge inizio(z1, i1) \wedge inizio(z2, i2) \wedge Fine(z1, f1) \wedge Fine(z2, f2)]$ $\rightarrow [f1 < i2 \vee f2 < i1]$ 9 Tipo: ~~Entità~~ | Relationship (cerchiare)Nome: Biglietto

attributo	dominio	moltepl. (*)
-----------	---------	--------------

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

Vincoli:

[V.biglietti_utenti] $\forall z [|\{u | ute_att(u, z) \vee \exists ac \text{ att.com}(z, ac) \wedge ute_att(u, ac)\}|$ $\neq 0] \rightarrow [|\{b | att_big(z, b)\}| = 0 \vee$ $|\{b | att_big(z, b)\}| =$ $|\{u | ute_att(u, z) \vee \exists ac \text{ att.com}(z, ac) \wedge ute_att(u, ac)\}|]$ 8 Tipo: ~~Entità~~ | Relationship (cerchiare)Nome: Biglietto

attributo	dominio	moltepl. (*)
-----------	---------	--------------

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

Vincoli:

[V.biglietti_utenti2] $\forall z$ $[|\{u | ute_att(u, z) \vee \exists ac \text{ att.com}(z, ac) \wedge ute_att(u, ac)\}| = 0]$ $\rightarrow [|\{b | att_big(z, b)\}| = 0 \vee$ $|\{b | att_big(z, b)\}| =$ $|\{u | ute_att(u, z) \vee \exists ac \text{ att.com}(z, ac) \wedge ute_att(u, ac)\}|]$

10 Tipo: Entità | Relationship (cerchiare)

Nome:

attributo	dominio	moltepl. (*)
-----------	---------	--------------

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

Vincoli:

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

OPERAZIONI DI CLASSE

AttComposta

prezzo_tot(): Reale ≥ 0

•pre-cond: nessuna

•post-cond:

$$A = \{(a, b) \mid \text{att_com}(a, \text{this}) \wedge \text{prezzo}(a, b)\}$$

$$\text{Result} = \sum_{(a, b) \in A} b$$

Attivita

fine(): DateTime

•pre-cond: nessuna

•post-cond: siano:

i tale che $\text{inizio}(\text{this}, i)$

d tale che $\text{durataMinuti}(\text{this}, d)$

Result = $i + d$ //somma DataOra + Minuti(intero) data dalla semantica del mondo reale

Viaggio

budget(): Reale ≥ 0

•pre-cond: nessuna

•post-cond:

$$A = \{(a, b, i) \mid \left[\left[\text{via_att}(\text{this}, a) \right] \vee \left[\exists zc \text{ AttComposta}(zc) \wedge \text{via_att}(zc, \text{this}) \right] \right] \wedge \text{att_com}(a, zc) \wedge \text{prezzo}(a, b) \wedge \text{inizio}(a, i) \}$$

$$\text{Result} = \sum_{(a, b, i) \in A} b$$

Tipi di Dato

Mail = [a-zA-Z0-9]+@[a-zA-Z0-9]+\.[a-z]{2,6}

Ind = (via: Stringa, civico: [0-9]+[a-z]{1,3})

Risposta alla Domanda 2 (segue)

Utente

punti(): intero ≥ 0

• pre_cond: $\exists k, v, u_2 \text{ crea}(\text{this}, v) \wedge \text{feedback}(u_2, v, k)$

• post_cond:

$$V = \{(u, k, v) \mid \text{crea}(\text{this}, v) \wedge \text{ute_via}(u, v) \wedge \text{Feedback}(u, v, k)\}$$

$$\text{media} = \sum_{(u, k, v) \in V} k \cdot \frac{1}{|V|}$$

$$V' = \{(u, k, v) \in V \mid k \geq 4\}$$

$$\text{point} = \lfloor |V'| \cdot 0.1 \rfloor$$

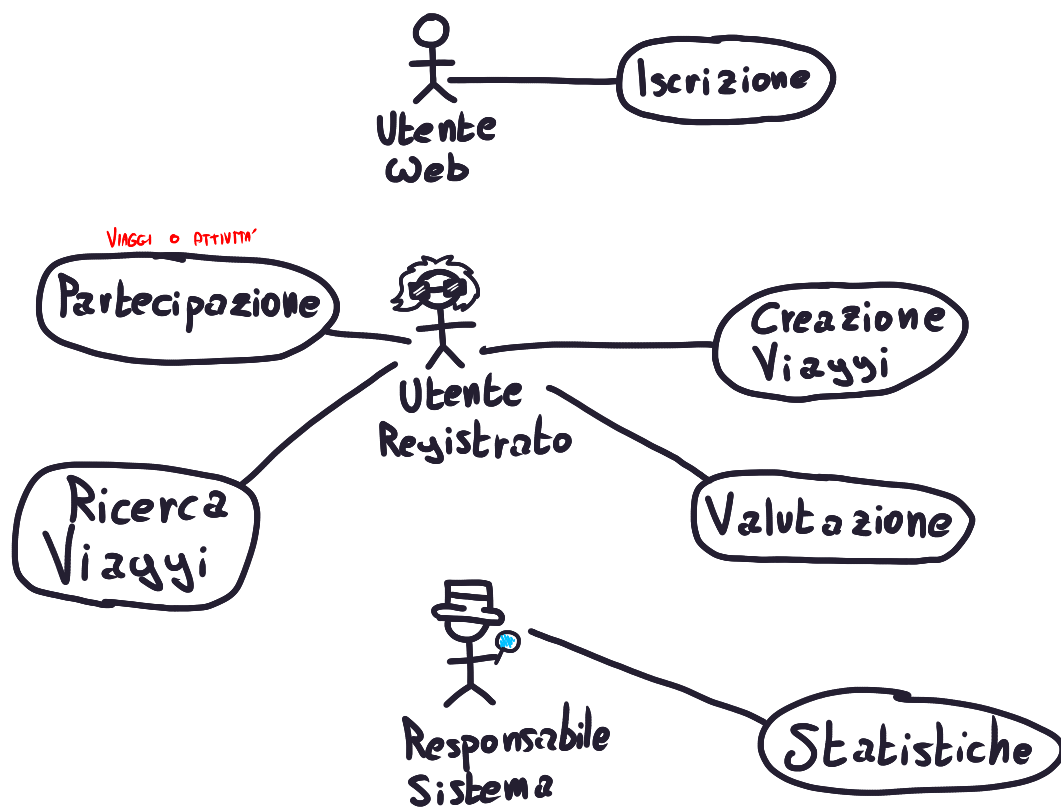
Result e' tale che:

$$\text{media} \leq 3 \rightarrow \text{Result} = 0 \wedge$$

$$\text{media} > 3 \rightarrow \text{Result} = \text{point}$$

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: Iscrizione (nome use-case)

Operazioni dello use-case:

iscriviti(nome:String, cognome:String, m:Mail):Utente

2 Specifica use-case: Creazione Viaggi (nome use-case)

Operazioni dello use-case:

crea(n:String, min:Intero>0, max:Intero>0, i:DataOra, f:DataOra, a:Attivita[0..], ac:AttComposta[0..*]):Viaggio*

aggiungi(v:Viaggio, a:Attivita[1..])*

3 Specifica use-case: Valutazione (nome use-case)

Operazioni dello use-case:

valuta(v:Viaggio, i:1..5)

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

Operazioni dello use-case:

partecipa-viaggio (*n*:Viaggio)

partecipa-attivita (*n*:Viaggio, *a*:Attivita)

partecipa-attivitaCon (*n*:Viaggio, *a*:AttComposta)

5 Specifica use-case: Statistiche (nome use-case)

Operazioni dello use-case:

viaggi_per_citta (*c*:Citta): (1..12, Intero ≥ 0) [12]

6 Specifica use-case: Ricerca Viaggi (nome use-case)

Operazioni dello use-case:

cerca_per_dest (*l*:luogo, *i*:DataOra, *f*:DataOra): Viaggio [0..*]

citta_viaggi (*i*:DataOra, *f*:DataOra): Citta [0..*]

ricerca-regione (*i*:DataOra, *f*:DataOra, *n*:Nazione): (Regione, Intero ≥ 0) [0..*]

cerca-budget-regioni (*bm*:Reale ≥ 0, *bM*:Reale ≥ 0, *R*:Regione [1..*], *i*:DataOra, *f*:DataOra, *p*:Intero ≥ 0): Viaggio [0..*]

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

verranno considerati gli usecase degni di nota

cerca-budget-regioni(bm:Reale ≥ 0 , bM:Reale ≥ 0 , R:Regione[1..*], i:DataOra, f:DataOra, p:Intero ≥ 0):Viaggio[0..*]

•pre-cond: $bm \leq bM \wedge i < f$

•post-cond:
$$V = \left\{ v \mid \begin{array}{l} \text{Viaggio}(v) \wedge \exists b \text{ budget}(v,b) \wedge bm \leq b \leq bM \wedge \exists i,v,fv \\ \text{inizio}(v,i) \wedge \text{fine}(v,fv) \wedge i \geq i \wedge fv \leq f \wedge \exists u \text{ crea}(u,v) \\ \wedge \exists K \text{ punti}(u,K) \wedge K \geq p \wedge \exists a [\text{via_att}(v,a) \vee [\exists zc \\ \text{via_attC}(v,zc) \wedge \text{att_com}(z,zc)]] \wedge \exists l,c,v \text{ att_lvo}(z,l) \\ \wedge \text{lvo_cit}(l,c) \wedge \text{cit_reg}(c,v) \wedge v \in R \end{array} \right\}$$

Result = V

viaggi_per_citta(c:Citta): (1..12, Interio ≥ 0)[12]

•pre-cond: nessuna

•post-cond: $V = \{ (m,n) \mid \text{viaggi_per_citta_mese}(c,m,n) \wedge 1 \leq m \leq 12 \wedge \text{Interio}(n) \}$

viaggi_per_citta_mese(c:Citta, m:1..12): Interio ≥ 0

•pre-cond: nessuna

•post-cond:
$$V = \left\{ v \mid \begin{array}{l} \text{Viaggio}(v) \wedge \exists i, an, mes \text{ inizio}(v,i) \wedge \text{Anno}(i, an) \\ \wedge \exists K \text{ Anno}(\text{ADESSO}, K) \wedge K = an \wedge \text{Mese}(i, mes) \wedge \\ mes = m \wedge \exists a, l \text{ Attivita}(a) \wedge \text{att_lvo}(a,l) \wedge \text{lvo_cit}(l,c) \\ \wedge [\text{via_att}(v,a) \vee [\exists zc \text{ via_attC}(v,zc) \wedge \text{att_com}(z,zc)]] \end{array} \right\}$$

Result = |V|

ADESSO è un simbolo di costante dato dalla semantica del mondo reale

Anno e **Mese** sono predicati dati dalla semantica del mondo reale

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 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 multivalore 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

Corrispondenza tra domini concettuali e domini supportati dal DBMS

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)

Risposta alla **Domanda 6** (segue)

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 (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 accorpa le relazioni che implementano le seguenti relationship:		

2	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 accorpa le relazioni che implementano le seguenti relationship:		

3	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 accorpa le relazioni che implementano le seguenti relationship:		

4	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 accorpa le relazioni che implementano le seguenti relationship:		

5	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 accorpa le relazioni che implementano le seguenti relationship:		

6 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 enupla, di dominio):

La relazione accorpa le relazioni che implementano le seguenti relationship:

7 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 enupla, di dominio):

La relazione accorpa le relazioni che implementano le seguenti relationship:

8 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 enupla, di dominio):

La relazione accorpa 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 enupla, di dominio):

La relazione accorpa 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 enupla, di dominio):

La relazione accorpa le relazioni che implementano le seguenti relationship:

11 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 enupla, di dominio):

La relazione accorpa le relazioni che implementano le seguenti relationship:

12 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 enupla, di dominio):

La relazione accorpa le relazioni che implementano le seguenti relationship:

13 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 enupla, di dominio):

La relazione accorpa le relazioni che implementano le seguenti relationship:

14 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 enupla, di dominio):

La relazione accorpa le relazioni che implementano le seguenti relationship:

15 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 enupla, di dominio):

La relazione accorpa le relazioni che implementano le seguenti relationship:

16 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 accorpa le relazioni che implementano le seguenti relationship:

17 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 accorpa le relazioni che implementano le seguenti relationship:

18 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 accorpa le relazioni che implementano le seguenti relationship:

19 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 accorpa le relazioni che implementano le seguenti relationship:

20 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 accorpa 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, enunpla, 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.

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 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

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.]

