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: .....

[Versione 2024-03-29]

Questo modulo è ottimizzato per la stampa fronte-retro

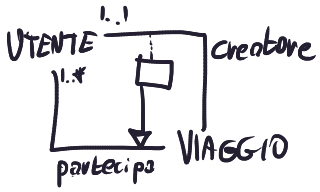
1 Analisi concettuale

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 → e Feedback (1..5)
   - 2.6 attività → 3. Attività
   - 2.7 budget() somma prezzi delle attività
   - 2.8 inizio()
   - 2.9 Fine()



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? {disj}
     - 3.9.2 Spostamento
       - 3.9.2.2 luogo arrivo
       - 3.9.2.1 mezzo
     - 3.9.3 Prenotazione

4. Attività composte
   - 4.1 insieme di attività
   - 4.2 prezzo_tot()

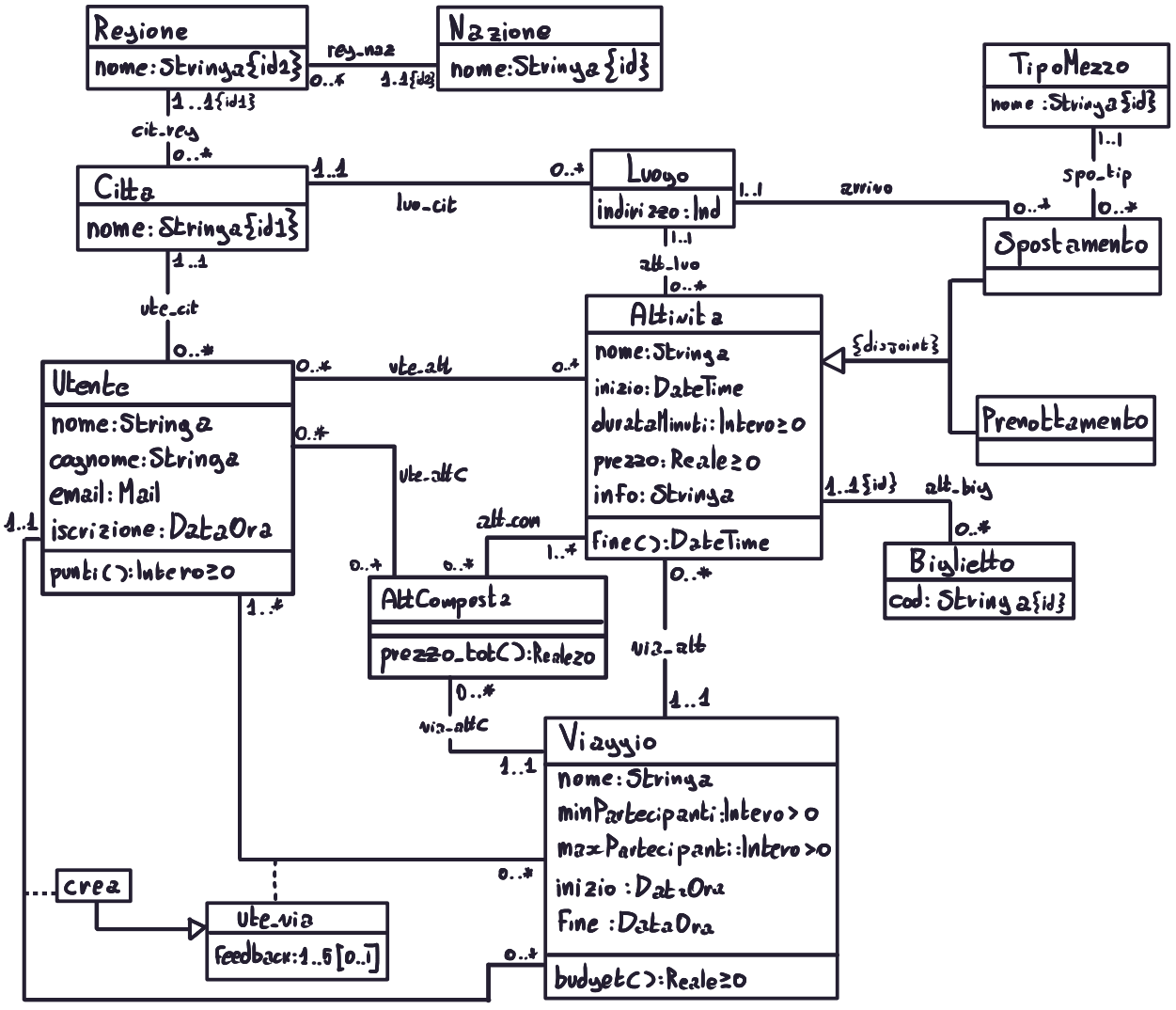
[continua alla pagina seguente]

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.



[continua alla pagina seguente]

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 $r(x_1, \ldots, x_n)$: $x_1, \ldots, x_n$ 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 $a(x_1, \ldots, x_n, v)$: uno dei valori dell'attr. a dell'istanza $(x_1, \ldots, x_n)$ 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

1 Tipo: (Entità) | **Relationship** (cerchiare)

Nome: Utente

attributo	dominio	moltepl. (*)

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

Vincoli:

[V.no-2-viaggi]

$\forall u, v1, v2, i1, i2, F1, F2$
$[Utente(u) \wedge Viaggio(v1) \wedge Viaggio(v2) \wedge vute\_via(u, v1) \wedge v1 \neq v2$
$vute\_via(u, v2) \wedge inizio(v1, i1) \wedge inizio(v2, i2) \wedge fine(v1, F1)$
$\wedge fine(v2, F2)] \rightarrow [F1 < i2 \vee F2 < i2]$

[V.iscritto_poi_viaggia]

$\forall u, v, i, n$
$[Utente(u) \wedge Viaggio(v) \wedge iscrizione(u, n) \wedge inizio(v, i)$
$\wedge vute\_via(u, v)] \rightarrow n < i$

2 Tipo: (Entità) | **Relationship** (cerchiare)

Nome: Utente

attributo	dominio	moltepl. (*)

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

Vincoli:

[V.no_auto_valutazioni]

$\forall u, v$
$[Utente(u) \wedge crea(u, v)] \rightarrow [\neg \exists F \ feedback(u, v, F)]$

[V.partecipa_att_del_viaggio] *

$\forall u, a\ [Utente(u) \wedge ute\_att(a, u)] \rightarrow [\exists v\ Viaggio(v)$
$vute\_via(u, v) \wedge [via\_att(v, a) \vee [\exists ac\ via\_attC(v, ac)$
$\wedge att\_com(a, ac)]]]$

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] *

$\forall v, m\,[Viaggio(v) \wedge maxPartecipanti(v,m)] \rightarrow |\{u \mid utente\_via(u,v)\}| \leq m$

5 Tipo: ~~Entità~~ | **Relationship** (cerchiare)

Nome: Prenotta mento

attributo	dominio	moltepl. (*)

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

Vincoli:

[V.no_stessa_data]

$\forall u, p_1, p_2, i_1, i_2, d_1, d_2$
$[p_1 \neq p_2 \wedge Utente(u) \wedge Prenotta mento(p_1) \wedge Prenotta mento(p_2) \wedge ute\_att(u,p_1) \wedge ute\_att(u,p_2) \wedge inizio(p_1,i_1) \wedge inizio(p_2,i_2) \wedge Data(i_1,d_1) \wedge Data(i_2,d_2)] \rightarrow d_1 \neq d_2$

4 Tipo: ~~Entità~~ | **Relationship** (cerchiare)

Nome: Viaggio

attributo	dominio	moltepl. (*)

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

Vincoli:

[V.attività_dentro_tempi_viaggio] *

$\forall v, iv, fv, z, iz, fz$
$[Viaggio(v) \wedge inizio(v,iv) \wedge Fine(v,iv) \wedge Attività(z) \wedge inizio(z,iz) \wedge Fine(z,fz) \wedge [via\_att(v,z) \vee [\exists zc\ via\_attC(v,zc) \wedge att\_com(z,zc)]]] \rightarrow [iv \leq iz \wedge fv \geq fz]$

6 Tipo: ~~Entità~~ | **Relationship** (cerchiare)

Nome: AttComposta

attributo	dominio	moltepl. (*)

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

Vincoli:

[V.inclusione_in_viaggio] *

$\forall z, zc, v$
$[Viaggio(v) \wedge AttComposta(zc) \wedge via\_attC(v,zc) \wedge att\_com(zc,z)] \rightarrow via\_att(v,z)$

[Versione 2024-03-29]

7 Tipo: ~~Entità~~ | **Relationship** (cerchiare)

Nome: Viaggio

attributo	dominio	moltepl. (*)

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

Vincoli:

[V. no_intersezione_attività]

$\forall v, z1, z2, i1, i2, F1, F2$
$[Viaggio(v) \wedge Attività(z1) \wedge Attività(z2) \wedge z1 \neq z2$
$\wedge inizio(z1, i1) \wedge inizio(z2, i2) \wedge Fine(z1, F1) \wedge Fine(z2, F2)]$
$\rightarrow [F1 < i2 \vee F2 < i1]$

9 Tipo: ~~Entità~~ | **Relationship** (cerchiare)

Nome: Biglietto

attributo	dominio	moltepl. (*)

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

Vincoli:

[V. biglietti_utenti]

$\forall z [|\{u | ute\_att(u,z) \vee \exists ac \ att\_com(z,ac) \wedge ute\_attc(u,ac)\}|$
$\neq 0] \rightarrow [|\{b | zlt\_big(z,b)\}| = 0 \vee$
$|\{b | zlt\_big(z,b)\}| =$
$|\{u | ute\_att(u,z) \vee \exists ac \ att\_com(z,ac) \wedge ute\_attc(u,ac)\}|]$

8 Tipo: ~~Entità~~ | **Relationship** (cerchiare)

Nome: Biglietto

attributo	dominio	moltepl. (*)

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

Vincoli:

[V. biglietti_utenti2]

$\forall z$

$[|\{u | ute\_att(u,z) \vee \exists ac \ att\_com(z,ac) \wedge ute\_attc(u,ac)\}| = 0]$

$\rightarrow [|\{b | zlt\_big(z,b)\}| = 0 \vee$

$|\{b | zlt\_big(z,b)\}| =$

$|\{u | ute\_att(u,z) \vee \exists ac \ att\_com(z,ac) \wedge ute\_attc(u,ac)\}|]$

10 Tipo: **Entità** | **Relationship** (cerchiare)

Nome: .....

attributo	dominio	moltepl. (*)

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

Vincoli:

[Versione 2024-03-29]

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:

$$A = \{(a,b) \mid att\_com(a, this) \wedge prezzo(a,b)\}$$

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

Attività

Fine(): DateTime
- pre-cond: nessuna
- post-cond: siano:
  i tale che inizio(this, i)
  d tale che durataMinuti(this, d)

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

Viaggio

budget(): Reale ≥ 0
- pre-cond: nessuna
- post-cond:

$$A = \left\{(a,b,i) \;\middle|\; [[via\_att(this,a)] \vee [\exists\, ac \;\; AttComposta(ac) \wedge via\_attC(this, ac) \wedge att\_com(a, ac)]] \wedge prezzo(a,b) \wedge inizio(a,i) \right\}$$

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

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})

[Versione 2024-03-29]

[continua alla pagina seguente]

Risposta alla Domanda 2 (segue)

Utente

punti(): Intero ≥ 0

- pre_cond: $\exists k, v, u_2$ crea(this, v) ∧ feedback(u_2, v, k)
- post_cond:

$$V = \{(u, k, v) \mid crea(this, v) \wedge ute\_via(u, v) \wedge feedback(u, v, k)\}$$

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

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

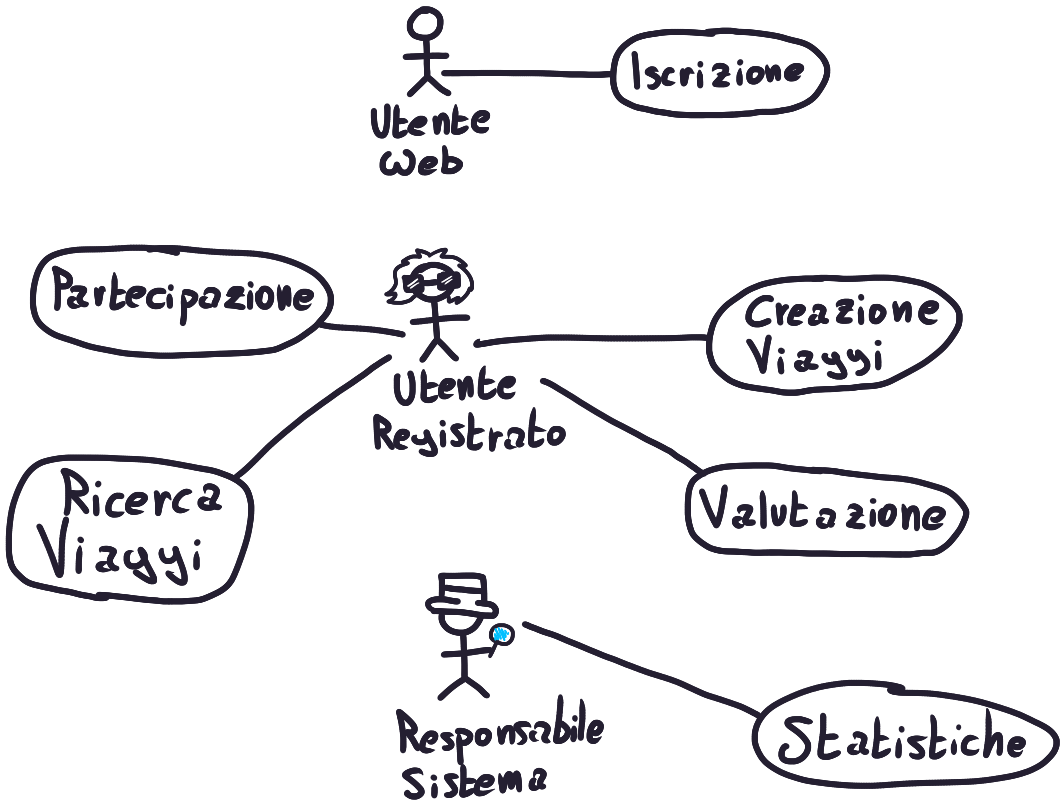
$$point = \lfloor |V'| \cdot 0.1 \rfloor$$

Result è tale che:

media ≤ 3 → Result = 0 ∧
media > 3 → Result = 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:Stringa, cognome:Stringa, m:Mail):Utente

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

Operazioni dello use-case:

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

aggiungi(v:Viaggio, a:Attività[1..*])

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

Operazioni dello use-case:

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

[Versione 2024-03-29]

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

Operazioni dello use-case:

partecipa_viaggio(v:Viaggio)

partecipa_attivita(v:Viaggio, a:Attivita)

partecipa_attivitaCon(v: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:

[Versione 2024-03-29]

[continua a pagina ..... delle minute]

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: $V = \{ v \mid Viaggio(v) \wedge \exists b\ budget(v,b) \wedge bm \leq b \leq bM \wedge \exists iv, fv\ inizio(v,iv) \wedge fine(v,fv) \wedge iv \geq i \wedge fv \leq F \wedge \exists u\ crea(u,v) \wedge \exists K\ punti(u,K) \wedge K \geq p \wedge \exists a [via\_att(v,a) \vee [\exists ac\ via\_attC(v,ac) \wedge att\_com(a,ac)]] \wedge \exists l,c,r\ att\_luo(a,l) \wedge luo\_cit(l,c) \wedge cit\_reg(c,r) \wedge r \in R \}$

  Result = V

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

- pre_cond: nessuna
- post_cond: $V = \{ (m,n) \mid viaggi\_per\_citta\_mese(c,m,n) \wedge 1 \leq m \leq 12 \wedge Intero(m) \}$

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

- pre_cond: nessuna
- post_cond: $V = \{ v \mid Viaggio(v) \wedge \exists i, an, mes\ inizio(v,i) \wedge Anno(i,an) \wedge \exists K\ Anno(ADESSO, K) \wedge K = an \wedge Mese(i,mes) \wedge mes = m \wedge \exists a,l\ Attivita(a) \wedge att\_luo(a,l) \wedge luo\_cit(l,c) \wedge [via\_att(v,a) \vee [\exists ac\ via\_attC(v,ac) \wedge att\_com(a,ac)]] \}$

  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

[Versione 2024-03-29]

[continua alla pagina seguente]

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.

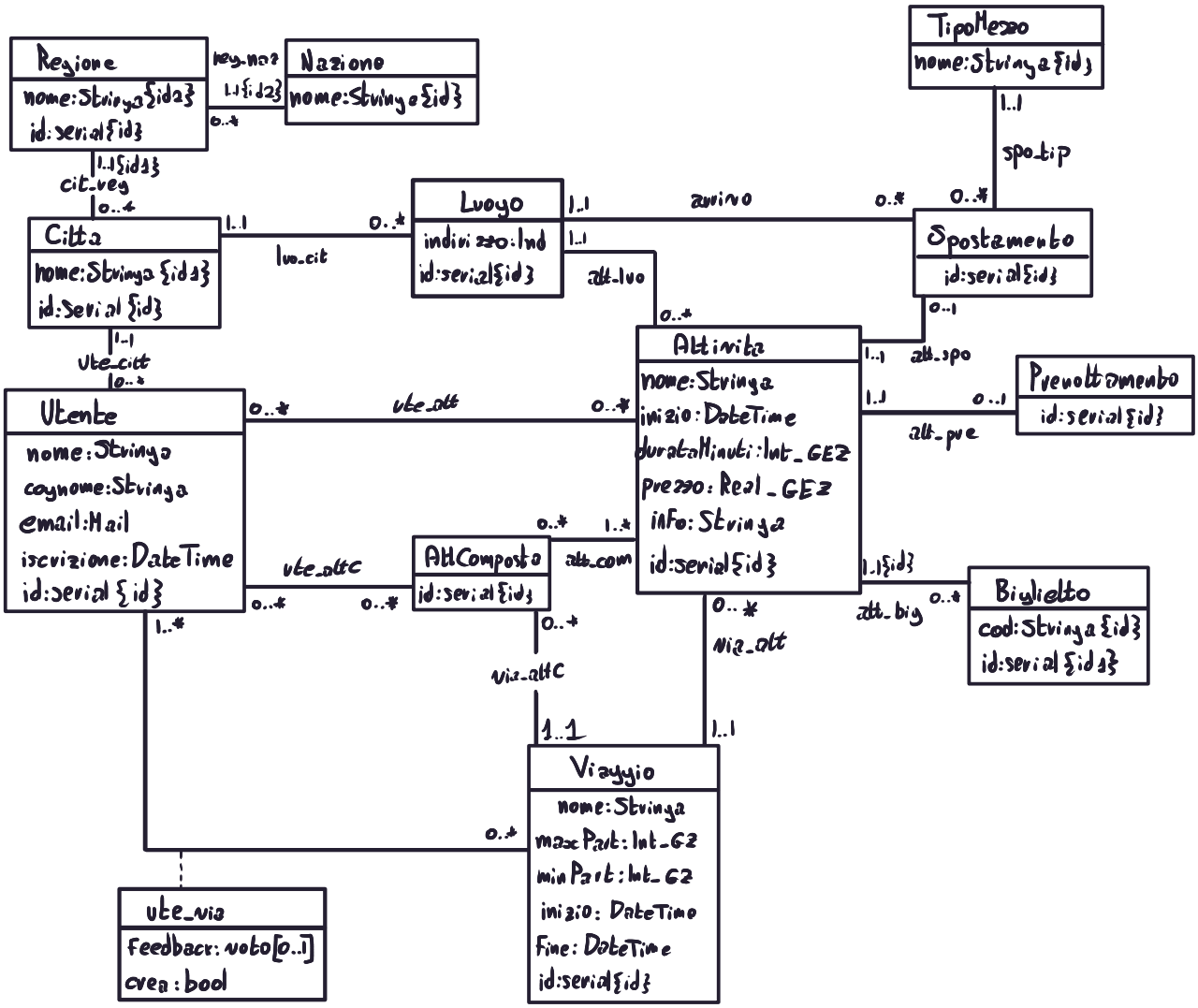
DBMS da utilizzare PostgreSQL.....

Corrispondenza tra domini concettuali e domini supportati dal DBMS

```
create domain Mail as varchar~'[a-zA-Z0-9]+@[a-zA-Z0-9]+.[a-z]{2,6}';
create domain Stringa as varchar NOT NULL;
create domain Civ as varchar~'[0-9]+[a-z]{1}';
create type ind as(
        via:Stringa
        civico: civ );
create domain Int_GZ as Integer check(value>0);
create domain Int_GEZ as Integer check( value >=0);
create domain Real_GEZ as Real check(value >=0);
create domain Voto as Integer check(value>=1 and value<=5);
```

[Versione 2024-03-29]

Diagramma ER ristrutturato



[Versione 2024-03-29]

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)

$\forall k, u, v \ [\text{feedback}(u, v, k)] \rightarrow [\exists f \ \text{crea}(u, v, f) \wedge f = \text{'False'}]$

$\forall a \ [\exists p \ \text{att\_pre}(a, p) \wedge \text{Attività}(a)] \rightarrow [\neg \exists s \ \text{att\_spo}(a, s)]$

$\forall a \ [\exists s \ \text{att\_spo}(a, s) \wedge \text{Spostamento}(s)] \rightarrow [\neg \exists p \ \text{att\_pre}(a, p)]$

[Versione 2024-03-29]

[continua alla pagina seguente]

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** Nazione ..... (nome) Derivante da: (entità) | **relationship** (cerchiare)

Attributi	nome						
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 accorpa le relazioni che implementano le seguenti relationship: .....

2 **Relazione** Regione ..... (nome) Derivante da: (entità) | **relationship** (cerchiare)

Attributi	nome	id	nazione				
Domini	Stringa	serial	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 nazione ref Nazione(nome);

unique(nome, nazione);

La relazione accorpa le relazioni che implementano le seguenti relationship: reg_naz .....

3 **Relazione** Città ..... (nome) Derivante da: (entità) | **relationship** (cerchiare)

Attributi	nome	id	regione				
Domini	Stringa	serial	Integer				

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 regione ref Regione(id);

Unique(nome, regione);

La relazione accorpa le relazioni che implementano le seguenti relationship: cit_reg .....

4 **Relazione** Utente ..... (nome) Derivante da: (entità) | **relationship** (cerchiare)

Attributi	nome	cognome	email	iscrizione	id	citta	
Domini	Stringa	Stringa	Mail	DateTime	serial	Integer	

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 citta ref Citta(id);

La relazione accorpa le relazioni che implementano le seguenti relationship: ute_cit .....

5 **Relazione** Viaggio ..... (nome) Derivante da: (entità) | **relationship** (cerchiare)

Attributi	nome	maxPart	minPart	inizio	Fine	id	
Domini	Stringa	Int_GZ	int_GZ	DateTime	DateTime	serial	

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(minPart <= maxPart); check(inizio < Fine);

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

[Versione 2024-03-29]

6 **Relazione** Ute_Via ..... (nome) Derivante da: **entità** | ~~**relationship**~~ (cerchiare)

Attributi	utente	viaggio	Feedback*	crea				
Domini	Integer	Integer	voto	bool				

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((feedback IS NOT NULL AND crea='False') OR (crea='True'));

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

7 **Relazione** Luogo ..... (nome) Derivante da: ~~**entità**~~ | **relationship** (cerchiare)

Attributi	indirizzo	id	citta					
Domini	ind	serial	Integer					

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 citta ref Citta(id);

La relazione accorpa le relazioni che implementano le seguenti relationship: luo_cit .....

8 **Relazione** Attività ..... (nome) Derivante da: ~~**entità**~~ | **relationship** (cerchiare)

Attributi	nome	inizio	durataMinuti	prezzo	info	id	luogo	viaggio
Domini	Stringa	DateTime	Int_GE2	Real_GE2	Stringa	serial	Integer	Integer

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 luogo ref Luogo(id);

Fk viaggio ref Viaggio(id);

La relazione accorpa le relazioni che implementano le seguenti relationship: via_att, att_luo .....

9 **Relazione** ~~AttComposta~~ ..... (nome) Derivante da: ~~**entità**~~ | **relationship** (cerchiare)

Attributi	id	viaggio						
Domini	serial	Integer						

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 viaggio ref Viaggio(id);

La relazione accorpa le relazioni che implementano le seguenti relationship: via_attC .....

10 **Relazione** att_com ..... (nome) Derivante da: **entità** | ~~**relationship**~~ (cerchiare)

Attributi	attC	att						
Domini	Integer	Integer						

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 attC ref AttComposta(id);

Fk att ref Attività(id);

V.inclusione AttComposta(id) incluso in att_com(attC);

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

[Versione 2024-03-29]

11 **Relazione** ute_alt ..... (nome) Derivante da: **entità** | **relationship** (cerchiare)

Attributi	utente	alt						
Domini	Integer	Integer						

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 alt ref Altinita(id);

Fk Utente ref Utente(id);

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

12 **Relazione** ute_altC ..... (nome) Derivante da: **entità** | **relationship** (cerchiare)

Attributi	utente	altC						
Domini	Integer	Integer						

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 utente ref Utente(id);

Fk altC ref AltComposta(id);

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

13 **Relazione** Biglietto ..... (nome) Derivante da: ~~entità~~ | **relationship** (cerchiare)

Attributi	cod	id	alt					
Domini	Stringa	serial	Integer					

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 alt ref Altinita(id);

Unique(alt, cod);

La relazione accorpa le relazioni che implementano le seguenti relationship: ..... alt_big .....

14 **Relazione** TipoMezzo ..... (nome) Derivante da: ~~entità~~ | **relationship** (cerchiare)

Attributi	nome							
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 accorpa le relazioni che implementano le seguenti relationship: .....

15 **Relazione** Spostamento ..... (nome) Derivante da: ~~entità~~ | **relationship** (cerchiare)

Attributi	id	luogo	mezzo	alt				
Domini	serial	Integer	Stringa	Integer				

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 luogo ref Luogo(id);

Fk mezzo ref TipoMezzo(nome); Fk alt ref Altinita(id); unique(alt);

La relazione accorpa le relazioni che implementano le seguenti relationship: alt_spo, spo_big, arrivo

[Versione 2024-03-29]

16 **Relazione** Prenotazione.....(nome) Derivante da: ~~entità~~ | **relationship** (cerchiare)

Attributi	alt	id						
Domini	Integer	Serial						

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):

unique(alt); fk alt ref Altività(id);

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: .....

[Versione 2024-03-29]

[continua a pagina ..... delle minute]

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.

da Fare i vincoli *

[continua alla pagina seguente]

Risposta alla Domanda 7 (segue) ogni trigger sarà controllato POST operazione

T.no_2_viaggi
Insert o Update ute_via

```
Error = EXISTS( SELECT *
                FROM ute_via uv, Viaggio v1, Viaggio v2
                WHERE uv.utente = new.utente
                AND v1 = uv.viaggio
                AND v2 = new.viaggio AND v1 <> v2
                AND (v1.inizio, v1.fine ) INTERSECT (v2.inizio, v2.fine));
```

if Error : rollback
else: permetti op.

T.iscritto_poi_viaggia
Insert o Update ute_via

```
Error = EXISTS ( SELECT *
                 FROM Utente u, viaggio v
                 WHERE new.utente = u.id
                 AND new.viaggio = v.id
                 AND v.inizio <= u.iscrizione);
```

if Error : rollback
else: permetti op.

T.no_intersezione_altivita
insert o Update Attivita

```
Error = EXISTS ( SELECT *
                 FROM Attivita a WHERE a.viaggio = new.viaggio
                 AND (a.inizio, fine_att(a.id)) INTERSECT (new.inizio, fine_att(new.id)));
```

if Error : rollback
else: permetti op.

T.no_stessa_data
Insert o Update ute_att

```
Error = EXISTS ( SELECT *
                 FROM ute_att at, Attivita a1, Prenotamento p, Attivita a2, Prenotamento p2
                 WHERE at <> new AND a2.id = new.att AND p2.att = a2.id
                 AND at.utente = new.utente AND at.att = a1.id AND a1.id = p.att
                 AND (a1.inizio, fine_att(a1.id)) INTERSET (a2.inizio, fine_att(a2.id)));
```

if Error : rollback
else: permetti op.

[Versione 2024-03-29]

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

Tabella di Regioni

cerca_budget_regioni(bm:Real_GE2, bM:Real_GE2, R:Insieme(<Integer>), i:DateTime, f:DateTime, p:Int_GE2) :Insieme(<Integer>)

- pre-cond: bm<=bM AND i<F

- post-cond:

```
V= SELECT v.id
   FROM Viaggio v, Attività a, Luogo l, Città c, R, vte_via
   WHERE a.viaggio=v.id AND a.luogo=l.id ∧ l.città=c.id AND città.regione=R.id
   AND v.inizio >=i AND v.fine <=F
   AND vte_via.viaggio=v.id AND vte_via.crea='True'
   AND punti(vte_via.utente)>=p
   AND budget(v.id)>=bm AND budget(v.id)<=bM;

result=V
```

```
viaggi_per_città(c: Integer)

    V= SELECT EXTRACT(MONTH FROM v.inizio), count(DISTINCT v.id)
       FROM Viaggio v, Attività a, Luogo l
       WHERE a.viaggio=v.id AND a.luogo=l.id
       AND l.città=c AND
       EXTRACT(YEAR FROM v.inizio)=EXTRACT(YEAR FROM NOW())
       GROUP BY EXTRACT(MONTH FROM v.inizio);

    Result=V
```

[Versione 2024-03-29]

[continua alla pagina seguente]

Risposta alla Domanda 8 (segue)

```
create function

fine_att(at:Integer):DateTime

        f=(SELECT inizio+INTERVAL durataMinuti::Interval
           FROM Attivita WHERE id=at.id);

        return f


Create function

budget(v:id):Real_GEZ

        b= SELECT sum(at.prezzo)
           FROM Attivita at
           WHERE at.viaggio=v;

        return b


create function

punti(u:Integer):Int_GEZ

        media= WITH Q as (SELECT v.id
                          FROM vte_via JOIN Viaggio v on v.id=vte_via.viaggio
                          AND crea='True' AND vte_via.utente=u)
               SELECT AVG(feedback)
               FROM vte_via JOIN Q on Q.v.id=vte_via.viaggio
               AND crea='False' AND feedback is NOT NULL;

         punti= WITH Q as (SELECT v.id
                          FROM vte_via JOIN Viaggio v on v.id=vte_via.viaggio
                          AND crea='True' AND vte_via.utente=u)
               SELECT count(*)
               FROM vte_via JOIN Q on Q.v.id=vte_via.viaggio
               AND crea='False' AND feedback > 3;

        if media <=3 :
            return 0
        else
            return Floor(punti . 0.1)
```

[Versione 2024-03-29]

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.]