



Basi di Dati, Modulo 2

Sapienza Università di Roma

Facoltà di Ing. dell'Informazione, Informatica e Statistica

Laurea in Informatica

Prof. Toni Mancini

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

Progetto 20220608 (P.20220608)

TravelPlan

Versione 2022-05-25

2

Specifiche dei Requisiti

Gli utenti di *TravelPlan* utilizzano il sistema per organizzare viaggi ed unirsi a viaggi organizzati da altri utenti.

Di ogni **utente** interessa conoscere il **nome, il cognome, l'indirizzo email, la città di provenienza e la data di iscrizione**.

Ogni utente può creare **viaggi** in qualità di **organizzatore**; dei viaggi interessa **il nome, il numero minimo e il numero massimo di partecipanti**. Gli utenti di *TravelPlan* possono quindi prendere parte ai viaggi organizzati dagli altri utenti.

Un **viaggio** è associato ad un certo numero di **attività** quali **visite, pasti, tour, trasporti e pernottamenti**. Di ogni **attività** interessa conoscere **il nome, l'istante di inizio, la durata, il prezzo, il luogo in cui si tiene, delle informazioni testuali ed eventualmente un insieme di codici identificativi di biglietti/prenotazioni relativi all'attività**.

Di ogni **luogo** interessa conoscere **l'indirizzo, la città, la regione e la nazione**.

Per facilitare la pianificazione, il sistema deve permettere di creare **attività composte**, cioè che raggruppano più **attività semplici**. Ad esempio, l'attività "tour gastronomico del centro storico" includerà la visita di due piazze, un percorso a piedi, un pasto al ristorante e la visita di una cantina.

Il sistema deve poter rappresentare gli **spostamenti** (con qualsiasi mezzo: treno, aereo, nave, bicicletta, a piedi, ecc.) **come attività**, delle quali è di interesse conoscere sia il **luogo di partenza che quello di arrivo**.

I **pernottamenti** sono **attività particolari, poiché il sistema non deve poter rappresentare più di un pernottamento al giorno per ogni utente che partecipa a un viaggio**.

Poiché può succedere che gli utenti che partecipano a uno stesso viaggio non prendano tutti parte alle stesse attività (p.es., due **sotto-gruppi** possono alloggiare in hotel diversi), il sistema deve permettere di specificare, per ogni attività inclusa in un viaggio, gli utenti che vi partecipano. **Se per una certa attività non è specificato nessun utente, si assume che tutti i partecipanti al viaggio vi prendano parte**.



I partecipanti ad un viaggio possono dare un **feedback** all'organizzazione del viaggio, assegnando un **voto da 1 a 5**.

TravelPlan prevede un **sistema di punteggi** per i propri utenti, in base alle valutazioni ricevute sui viaggi da loro organizzati. **Il punteggio p di un utente è così calcolato: p è uguale a 0 se la media aritmetica dei voti ricevuti è ≤ 3 , altrimenti p è uguale alla parte intera inferiore del 10% del numero dei feedback ricevuti con voto ≥ 4 .**

Il sistema deve permettere agli utenti registrati di partecipare a viaggi e di creare viaggi in qualità di organizzatori, aggiungendo a essi attività di vario tipo.

Infine, il sistema deve offrire le seguenti funzionalità avanzate:

1. Gli utenti registrati devono poter:
 - 1.1. **creare viaggi** in qualità di organizzatori e aggiungere a essi attività di vario tipo;
 - 1.2. **trovare tutti i viaggi** che includono una certa destinazione in un certo intervallo di tempo;
 - 1.3. **trovare la/e città** toccata/e dal maggior numero di viaggi in un certo intervallo di tempo;
 - 1.4. **calcolare, per ogni regione di una data nazione, il numero di viaggi organizzati in un dato periodo di tempo che toccano quella regione;**
 - 1.5. **dato un budget minimo, un budget massimo, un insieme di regioni, un periodo di tempo e un punteggio**, calcolare l'insieme dei viaggi che hanno un budget nell'intervallo richiesto, toccano almeno una delle regioni date, si svolgono completamente all'interno del periodo dato e sono organizzati da un utente con un punteggio pari o superiore a quello dato.
2. I responsabili del sistema devono poter calcolare, data una città di destinazione, per ognuno dei 12 mesi dell'anno, il **numero di viaggi organizzati in quel mese nell'ultimo anno solare.**



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

Esame BD2.Esame.Risposte – Modulo risposte prova scritta

Dati dello studente e dell'esame

Cognome e nome: BIANCO SIMONE Matricola: 11

Data: 19/08/23

Corso di laurea e canale di appartenenza:

- Laurea in Informatica, canale 1 (A-L, Prof. M. Mancini)
- 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:

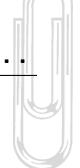
.....

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.



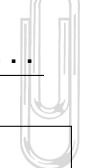
Questa pagina è stata intenzionalmente lasciata vuota

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

UTENTE	VIAGGIO	ATTIVITA'
<ul style="list-style-type: none"> • NAME, COGNOME • EMAIL • CITTÀ • ISCRIZIONE • CREA E PARTECIPA A VIAGGI 	<ul style="list-style-type: none"> • COMPOSTO DA ATTIVITA' • INIZIO • FINE • MIN e MAX • PARTECIPANTI 	<ul style="list-style-type: none"> • INIZIO • DURATA - MIN • NAME • PREFIZZO • LUOGO • INFO • CODICI (0, N) • VISITA, PASTO, tour, PERNOVAMENTO, SPOSTAMENTO
	LUOGO <ul style="list-style-type: none"> • INDIRIZZO • CITTÀ • REGION • NAZIONE 	
		FEEDBACK
<ul style="list-style-type: none"> • AL VIAGGIO • DA 1 A 5 	PERNOVAMENTO <ul style="list-style-type: none"> • SOLO UNO AL GIORNO PER UTENTE 	SPOSTAMENTI <ul style="list-style-type: none"> • ANCHE LUOGO ARRIVO



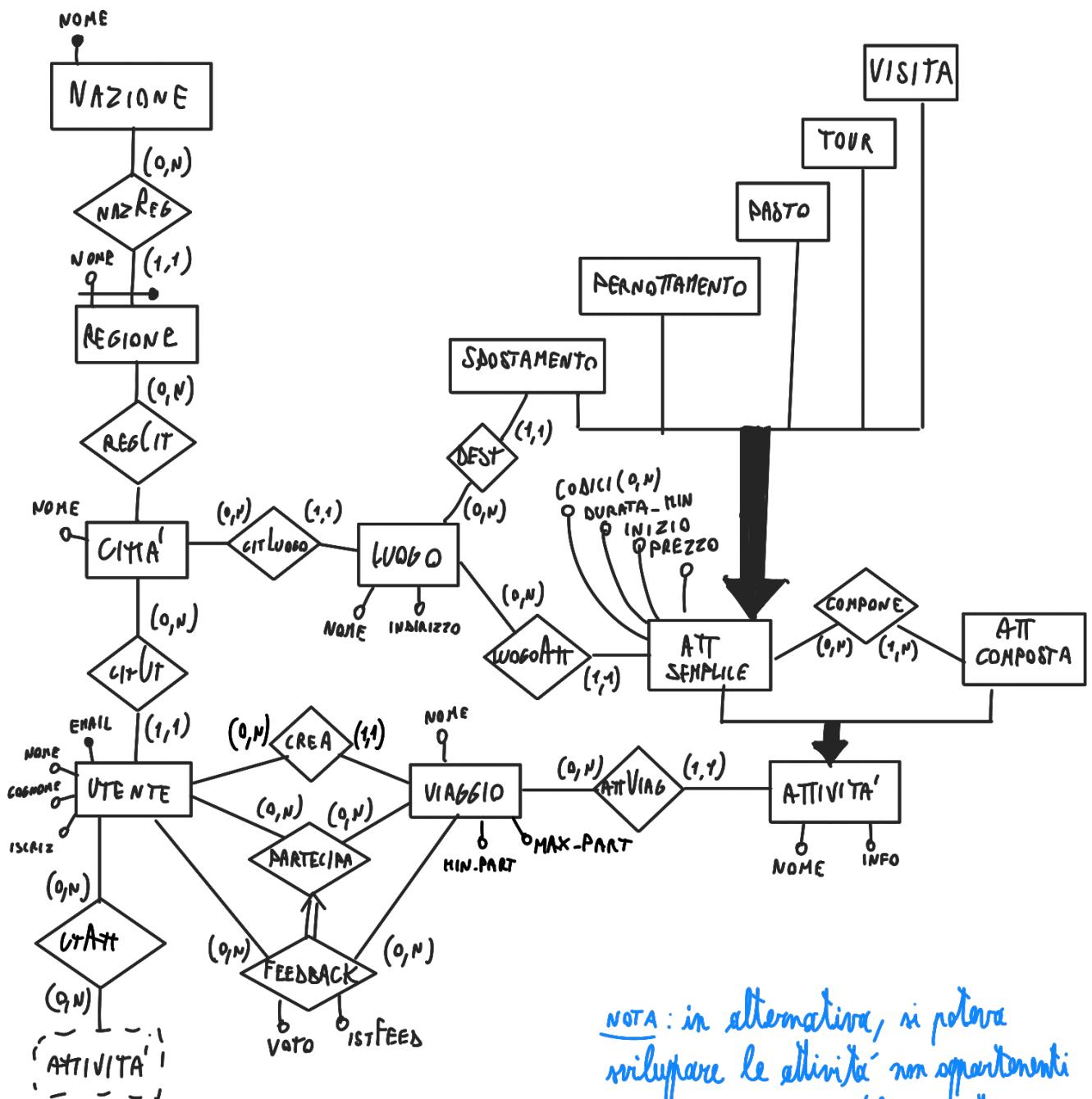
Risposta alla Domanda 1 (segue)

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.

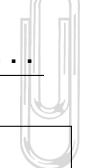
Una risposta soddisfacente a questa domanda è condizione *necessaria* (ma non sufficiente) per superare la prova.

Diagramma ER

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



NOTA: in alternativa, si poteva sviluppare le attivita' non appartenenti ad una componete come delle componete con una sola attivita' semplice.



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

<p>[1] Tipo: Entità Relationship (cerchiare)</p> <p>Nome: ...<u>NAZIONE</u>...</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;">moltepl. (*)</th></tr> </thead> <tbody> <tr> <td style="text-align: center;"><u>NOME</u></td><td style="text-align: center;"><u>STRINGA</u></td><td></td></tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p>	attributo	dominio	moltepl. (*)	<u>NOME</u>	<u>STRINGA</u>		<p>[2] Tipo: Entità Relationship (cerchiare)</p> <p>Nome: ...<u>REGIONE</u>...</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;">moltepl. (*)</th></tr> </thead> <tbody> <tr> <td style="text-align: center;"><u>NOME</u></td><td style="text-align: center;"><u>STRINGA</u></td><td></td></tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p>	attributo	dominio	moltepl. (*)	<u>NOME</u>	<u>STRINGA</u>	
attributo	dominio	moltepl. (*)											
<u>NOME</u>	<u>STRINGA</u>												
attributo	dominio	moltepl. (*)											
<u>NOME</u>	<u>STRINGA</u>												

3	Tipo: Entità Relationship (cerchiare)	
Nome:	CITTÀ'	
attributo	dominio	moltep. (*)
NOME	STRINGA	

5	Tipo: Entità Relationship (cerchiare)	
Nome:	UTENTE	
attributo	dominio	moltep. (*)
NOME	STRINGA	
COGNOME	STRINGA	
ISCRIZ	DATA	
EMAIL	EMAIL	

4	Tipo: Entità Relationship (cerchiare)	
Nome:	VIAGGIO	
attributo	dominio	moltep. (*)
NOME	STRINGA	
MIN PART	INTERO	
MAX PART	INTERO	

6	Tipo: Entità Relationship (cerchiare)	
Nome:	FEEDBACK	
attributo	dominio	moltep. (*)
VOTO	INTERO [1,5]	
IS FEED	DATA ORA	

<p>7 Tipo: Entità Relationship (cerchiare)</p> <p>Nome: ... ATTIVITÀ</p> <table border="1" data-bbox="96 242 790 303"> <thead> <tr> <th>attributo</th><th>dominio</th><th>moltep. (*)</th></tr> </thead> <tbody> <tr> <td>NOME</td><td>STRINGA</td><td></td></tr> <tr> <td>INFO</td><td>STRINGA</td><td></td></tr> </tbody> </table> <hr/> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p> <ul style="list-style-type: none"> • V. ATTIVITÀ. UTENTE $\forall u, v, a \ ATTIVITÀ(a) \wedge \text{UTEN}(a, u) \wedge$ $ATTIVAG(v, a) \rightarrow \text{PARTECIPA}(u, v)$ 	attributo	dominio	moltep. (*)	NOME	STRINGA		INFO	STRINGA		<p>9 Tipo: Entità Relationship (cerchiare)</p> <p>Nome: ... PERNOTTAMENTO</p> <table border="1" data-bbox="795 242 1505 303"> <thead> <tr> <th>attributo</th><th>dominio</th><th>moltep. (*)</th></tr> </thead> <tbody> <tr> <td></td><td></td><td></td></tr> </tbody> </table> <hr/> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p> <p>R • V. PERNOTTAMENTO. UNO AL GIORNO</p> <p>$\forall u, p, p', i, i', d, d' \ UTENTE(u) \wedge$ $\text{PERNOTTAMENTO}(p) \wedge \text{PERNOTTAMENTO}(p') \wedge p \neq p' \wedge$ $u \in \text{PARTECIPANTI}(p) \wedge u \notin \text{PARTECIPANTI}(p') \wedge$ $\text{INIZIO}(p, i) \wedge \text{DATA}(i, d) \wedge \text{INIZIO}(p', i') \wedge$ $\text{DATA}(i', d') \rightarrow d \neq d'$</p>	attributo	dominio	moltep. (*)			
attributo	dominio	moltep. (*)														
NOME	STRINGA															
INFO	STRINGA															
attributo	dominio	moltep. (*)														

<p>8 Tipo: Entità Relationship (cerchiare)</p> <p>Nome: ... LUOGO</p> <table border="1" data-bbox="88 1311 790 1417"> <thead> <tr> <th>attributo</th><th>dominio</th><th>moltep. (*)</th></tr> </thead> <tbody> <tr> <td>INDIRIZZO</td><td>INDIRIZZO</td><td></td></tr> <tr> <td>NOME</td><td>STRINGA</td><td></td></tr> </tbody> </table> <hr/> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p> <p>R • V. LUOGO. SPOSTAMENTO</p> <p>$\forall s, l, l' \ SPOSTAMENTO(s) \wedge \text{LUOGO_ATT}(s, l)$ $\rightarrow \exists \text{DEST}(s, l')$</p>	attributo	dominio	moltep. (*)	INDIRIZZO	INDIRIZZO		NOME	STRINGA		<p>10 Tipo: Entità Relationship (cerchiare)</p> <p>Nome: ... ATT COMPOSTA</p> <table border="1" data-bbox="795 1311 1505 1417"> <thead> <tr> <th>attributo</th><th>dominio</th><th>moltep. (*)</th></tr> </thead> <tbody> <tr> <td></td><td></td><td></td></tr> </tbody> </table> <hr/> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p> <p>R • V. ATT COMPOSTA. VIAGGIO</p> <p>$\forall ac, aa, v \ ATT\text{COMPOSTA}(ac) \wedge \text{COMPONE}(ac, aa) \wedge$ $ATTVIAGGIO(v, ac) \rightarrow ATTVIAGGIO(v, aa)$</p> <p>R • V. ATT COMPOSTA. UTENTE</p> <p>$\forall ac, aa, u \ ATT\text{COMPOSTA}(ac) \wedge \text{COMPONE}(ac, aa) \wedge$ $UTATT(ac, u) \rightarrow UTATT(aa, u)$</p>	attributo	dominio	moltep. (*)			
attributo	dominio	moltep. (*)														
INDIRIZZO	INDIRIZZO															
NOME	STRINGA															
attributo	dominio	moltep. (*)														

<p>11 Tipo: Entità Relationship (cerchiare)</p> <p>Nome: ... ATT_SEMPlice</p> <table border="1" data-bbox="96 249 790 473"> <thead> <tr> <th>attributo</th><th>dominio</th><th>moltep. (*)</th></tr> </thead> <tbody> <tr> <td>PREZZO</td><td>DENARO</td><td></td></tr> <tr> <td>CODICI</td><td>STR</td><td>(0,n)</td></tr> <tr> <td>INIZIO</td><td>DATAORA</td><td></td></tr> <tr> <td>DURATA_MIN</td><td>INT-62</td><td></td></tr> </tbody> </table>	attributo	dominio	moltep. (*)	PREZZO	DENARO		CODICI	STR	(0,n)	INIZIO	DATAORA		DURATA_MIN	INT-62		<p>13 Tipo: Entità Relationship (cerchiare)</p> <p>Nome:</p> <table border="1" data-bbox="795 249 1505 473"> <thead> <tr> <th>attributo</th><th>dominio</th><th>moltep. (*)</th></tr> </thead> </table>	attributo	dominio	moltep. (*)
attributo	dominio	moltep. (*)																	
PREZZO	DENARO																		
CODICI	STR	(0,n)																	
INIZIO	DATAORA																		
DURATA_MIN	INT-62																		
attributo	dominio	moltep. (*)																	
<p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p> <ul style="list-style-type: none"> • V. ATT_SEMPlice. DISOLVENTE $\begin{aligned} & \forall u, a, a', i, i', d, d' \quad UTENTE(u) \wedge ATT_SEMPlice(a) \wedge \\ & ATT_SEMPlice(a') \wedge a \neq a' \wedge INIZIO(a, i) \wedge INIZIO(a', i') \\ & \wedge DURATA_MIN(a, d) \wedge DURATA_MIN(a', d') \\ & \rightarrow \exists t \quad DATAORA(t) \wedge (i \leq t \wedge t \leq i + d \cdot '1 MINUTO') \\ & \wedge (i' \leq t \wedge t \leq i' + d' \cdot '1 MINUTO') \end{aligned}$	<p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p>																		

<p>12 Tipo: Entità Relationship (cerchiare)</p> <p>Nome:</p> <table border="1" data-bbox="87 1282 790 1327"> <thead> <tr> <th>attributo</th><th>dominio</th><th>moltep. (*)</th></tr> </thead> </table>	attributo	dominio	moltep. (*)	<p>14 Tipo: Entità Relationship (cerchiare)</p> <p>Nome:</p> <table border="1" data-bbox="795 1282 1505 1327"> <thead> <tr> <th>attributo</th><th>dominio</th><th>moltep. (*)</th></tr> </thead></table>	attributo	dominio	moltep. (*)
attributo	dominio	moltep. (*)					
attributo	dominio	moltep. (*)					
<p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p>	<p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p>						

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

Nome:

attributo	dominio	moltepl. (*)

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

Vincoli:

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

Nome:

attributo	dominio	moltepl. (*)

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

Vincoli:

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

Nome:

attributo	dominio	moltepl. (*)

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

Vincoli:

18 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

DOMINIO EMAIL: STRINGA SECONDO STANDARD

DOMINIO DENARO:

VALUTA: STRINGA 3 CARATTERI

IMPORTO: REALE > 0

DOMINIO INDIRIZZO:

VIA: STRINGA

CIVICO: INTERO > 0

CAP: STRINGA 5 CIFRE

PARTECIPANTI (a: ATTIVITA'): UTENTE (0, n)

PRE:

POST:

- NO SIDE EFFECTS

- VAL RITORNO

$$U_A = \{u \mid \text{UTENTE}(u) \wedge \text{ATT}(a, u)\}$$

$$U_V = \{u' \mid \text{UTENTE}(u') \wedge (\exists v \text{ VIAGGIO}(v) \wedge \text{ATT}(V, a) \wedge \text{PARTECIPA}(u, v))\}$$

$$(U_A \neq \emptyset \rightarrow \text{RESULT} = U_A) \wedge (U_A = \emptyset \rightarrow \text{RESULT} = U_V)$$

INIZIO AttComposta (a: AttComposta): DATAORA

PRE:

POST:

$$A = \{i \mid \exists a' \text{ ATTSIMPLICE}(a') \wedge \text{INIZIO}(a', i) \wedge \text{COMPONE}(a', a)\}$$

$$\text{RESULT} = \min(A)$$

FINE AttComposta (a: AttComposta): DATAORA

PRE:

POST:

$$A = \left\{ \begin{array}{l} \exists a', i, d \text{ ATTSIMPLICE}(a') \wedge \text{INIZIO}(a', i) \\ \wedge \text{DURATA_MIN}(a', d) \wedge f = i + d \cdot 1 \text{ MINUTO}' \\ \wedge \text{COMPONE}(a', a) \end{array} \right\}$$

$$\text{RESULT} = \max(A)$$

Risposta alla Domanda 2 (segue)

$\text{INIZIO VIAGGIO}(v : \text{VIAGGIO})$: DATAORA

PRE:

POST:

$$A = \left\{ i \mid \exists z \text{ ATTEMPICE}(z) \wedge \text{INIZIO}(z, i) \wedge \text{ATTVIAG}(v, z) \right\}$$

$$\text{RESULT} = \min(A)$$

$\text{FINE VIAGGIO}(v : \text{VIAGGIO})$: DATAORA

PRE:

POST:

$$A = \left\{ f \mid \begin{array}{l} \exists z, i, d \text{ ATTEMPICE}(z) \wedge \text{INIZIO}(z, i) \\ \wedge \text{DURATA_MIN}(z, d) \wedge f = i + d \cdot 1 \text{ MINUTO}' \\ \wedge \text{ATTVIAG}(v, z) \end{array} \right\}$$

$$\text{RESULT} = \max(A)$$

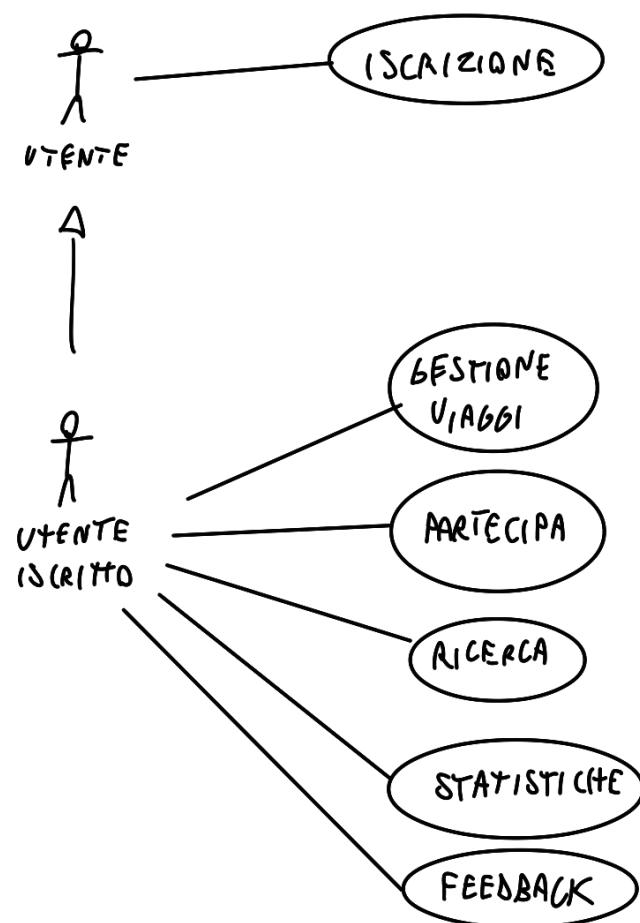
- $\text{V. VIAGGIO. PARTECIPANTI}$

$$\forall v, x, y \quad \text{VIAGGIO}(v) \wedge \text{MINPART}(v, x) \wedge \text{MAXPART}(v, y) \rightarrow x \leq \left| \left\{ u \mid \begin{array}{l} \text{UTENTE}(u) \wedge \\ \text{PARTECIPA}(u, v) \end{array} \right\} \right| \leq y$$

- $\text{V. VIAGGIO. MINMAX}$

$$\nexists v, x, y \quad \text{VIAGGIO}(v) \wedge \text{MINPART}(v, x) \wedge \text{MAXPART}(v, y) \rightarrow x \leq y$$

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: **(SCRIZIONE)** (nome use-case)

Operazioni dello use-case:

$\text{ISCRIVI } (\alpha: \text{STRINGA}, c: \text{STRINGA}, e: \text{EMAIL}, d: \text{DATA}, c': \text{CITTÀ}): \text{UTENTE}$

2 Specifica use-case: **GESTIONE VIAGGI** (nome use-case)

Operazioni dello use-case:

$\text{CREA VIAGGIO } (u: \text{UTENTE}, n: \text{STRINGA}): \text{VIAGGIO}$

$\text{CREA ATTIVITÀ SEMPLICE } (v: \text{VIAGGIO}, n: \text{STRINGA}, i: \text{DATAORA}, d: \text{INTERO} > 0, c': \text{STRINGA}, m: \text{DENARO}): \text{ATT SEMPLICE}$

$\text{CREA ATTIVITÀ COMPOSTA } (v: \text{VIAGGIO}, n: \text{STRINGA}, i: \text{STRINGA}, AS: \text{ATT SEMPLICE}, (1, N)): \text{ATT COMPOSTA}$

3 Specifica use-case: **PARTECIPAZIONE** (nome use-case)

Operazioni dello use-case:

$\text{PARTECIPA VIAGGIO } (u: \text{UTENTE}, v: \text{VIAGGIO})$

$\text{PARTECIPA ATTIVITÀ } (u: \text{UTENTE}, z: \text{ATTIVITÀ})$

$\text{PARTECIPANTI } (a: \text{ATTIVITÀ}): \text{UTENTE}(0, N)$

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

Operazioni dello use-case:

~~RICERCA VIAGGI LUOGO (l: luogo, i: DATAORA, f: DATAORA): VIAGGIO (0, n)~~

~~RICERCA CITTÀ POPOLARE (i: DATAORA, f: DATAORA): CITTÀ' (0, n)~~

RICERCA VIAGGI PARAMETRI (R: REGIONE (0, n), i: DATAORA, f: DATAORA, b_{min}: DENARO, b_{max}: DENARO, p: INTERO > 0): VIAGGIO (0, n)

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

Operazioni dello use-case:

~~CALCOLA PUNTEGGIO (a: UTENTE): INTERO > 0~~

NUMVIAGGI REGIONI NAZIONE (n: NAZIONE, i: DATAORA, f: DATAORA): (n: REGIONE, n': INTERO > 0) (0, n)

~~CALCOLA COSTO VIAGGIO (v: VIAGGIO): DENARO~~

NUMVIAGGI ULTIMO ANNO (c: CITTÀ'): (m: INTERO [1, 12], n: INTERO > 0) (0, n)

6 Specifica use-case: FEEDBACK (nome use-case)

Operazioni dello use-case:

LASCIA FEEDBACK (u: UTENTE, v: VIAGGIO, i: INTERO [1, 5])

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

CALCOLA PUNTEGGIO (u: UTENTE): INTERO ≥ 0

PRE:

POST:

$$V = \left\{ (u', v', v') \mid \begin{array}{l} UTENTE(u') \wedge VIAGGIO(v) \wedge CREA(u, v) \\ PARTECIPA(u', v) \wedge FEEDBACK(u', v) \wedge VOTO(u', v, v') \end{array} \right\}$$

$$(V = \emptyset \rightarrow m = 0) \wedge \left(V \neq \emptyset \rightarrow m = \frac{\sum_{(u', v', v') \in V} v'}{|V|} \right)$$

$$V_{\geq 4} = \left\{ (u', v', v') \mid (u', v', v') \in V \wedge v' \geq 4 \right\}$$

$$(m \leq 3 \rightarrow \text{RESULT} = 0) \wedge (m \geq 4 \rightarrow \text{RESULT} = \lfloor 0.1 \cdot |V_{\geq 4}| \rfloor)$$

CALCOLA COSTO VIAGGIO (v: VIAGGIO): DENARO

PRE:

POST:

$$A = \left\{ (a, p) \mid ATTIVITASEMPLICE(a) \wedge PREZZO(a, p) \wedge ATTIVITA(v, a) \right\}$$

$$\text{RESULT} = \sum_{(a, p) \in A} p$$

NUM VIAGGI REGIONI NAZIONE (n: NAZIONE, i: DAYMORA, k: DATAORA): ($\eta: REGIONE, \eta': INTERO \geq 0$) (0, n)

PRE: $i < k$

POST:

$$R = \left\{ (\eta, k) \mid \begin{array}{l} REGIONE(\eta) \wedge NAZREG(\eta, n) \wedge \\ K = \left\{ v \mid \begin{array}{l} VIAGGIO(v) \wedge i \leq INIZIOVIAGGIO(v) \wedge FINEVIAGGIO(v) \leq k \\ (\exists a, l, c \quad ATTIVITA'(a) \wedge ATTIVITAVIAG(v, a) \wedge (LUOGOATT(a, l) \\ \vee DEST(a, l)) \wedge circuitoso(l, c) \wedge REG(ET(c, n)) \end{array} \right\} \end{array} \right\}$$

$$\text{RESULT} = R$$

Risposta alla Domanda 5 (segue)

RICERCA VIAGGI PARAMETRI ($R: REGIONE(o,n)$, $i: DATAORA$, $f: DATAORA$, $b_{min}: DENARO$, $b_{max}: DENARO$, $P: INTERO>0$): VIAGGIO(o, n)
 PRE: $i < f \wedge b_{min} < b_{max}$

POST:

$$V = \left\{ v \mid \begin{array}{l} VIAGGIO(v) \wedge i \leq INIZIO(VIAGGIO(v)) \wedge FINE(VIAGGIO(v)) \leq f \\ (\exists u, a, l, c, r, k \text{ CREA}(u, v) \wedge CALCOLA(PUNTEGGIO(u), P) \geq P) \\ CALCOLA(COSTO(VIAGGIO(v)) = k \wedge b_{min} \leq k \wedge k \leq b_{max}) \wedge \\ \wedge ATT(VIAGGIO(v), a) \wedge (LUOGOATT(a, l) \vee DEST(a, l)) \wedge \\ \wedge CITLUOGO(l, c) \wedge REG(1T(c, r) \wedge r \in R) \end{array} \right\}$$

RESULT = V

NUMVIAGGIULTIMOANNO($c: CITTA'$): ($m: INTERO[1, 12]$, $n: INTERO>0$) (o, n)

PRE.

POST:

$$M = \left\{ (m, n) \mid \begin{array}{l} INTERO(m) \wedge m \geq 1 \wedge m \leq 12 \wedge \\ n = \left\{ v \mid \begin{array}{l} VIAGGIO(v) \wedge (\exists i, f, a, di, l \mid i = INIZIO(VIAGGIO(v)) \\ \wedge f = FINE(VIAGGIO(v)) \wedge (ADESSO - '1 ANNO') \leq i \\ \wedge f \leq ADESSO \wedge DATA(i, di) \wedge DATA(f, dl) \\ \wedge MESE(di, m) \wedge ATT(VIAG(v), a) \wedge \\ (LUOGOATT(a, l) \vee DEST(a, l)) \wedge CITLUOGO(l, c)) \end{array} \right\} \end{array} \right\}$$

RESULT = M

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**

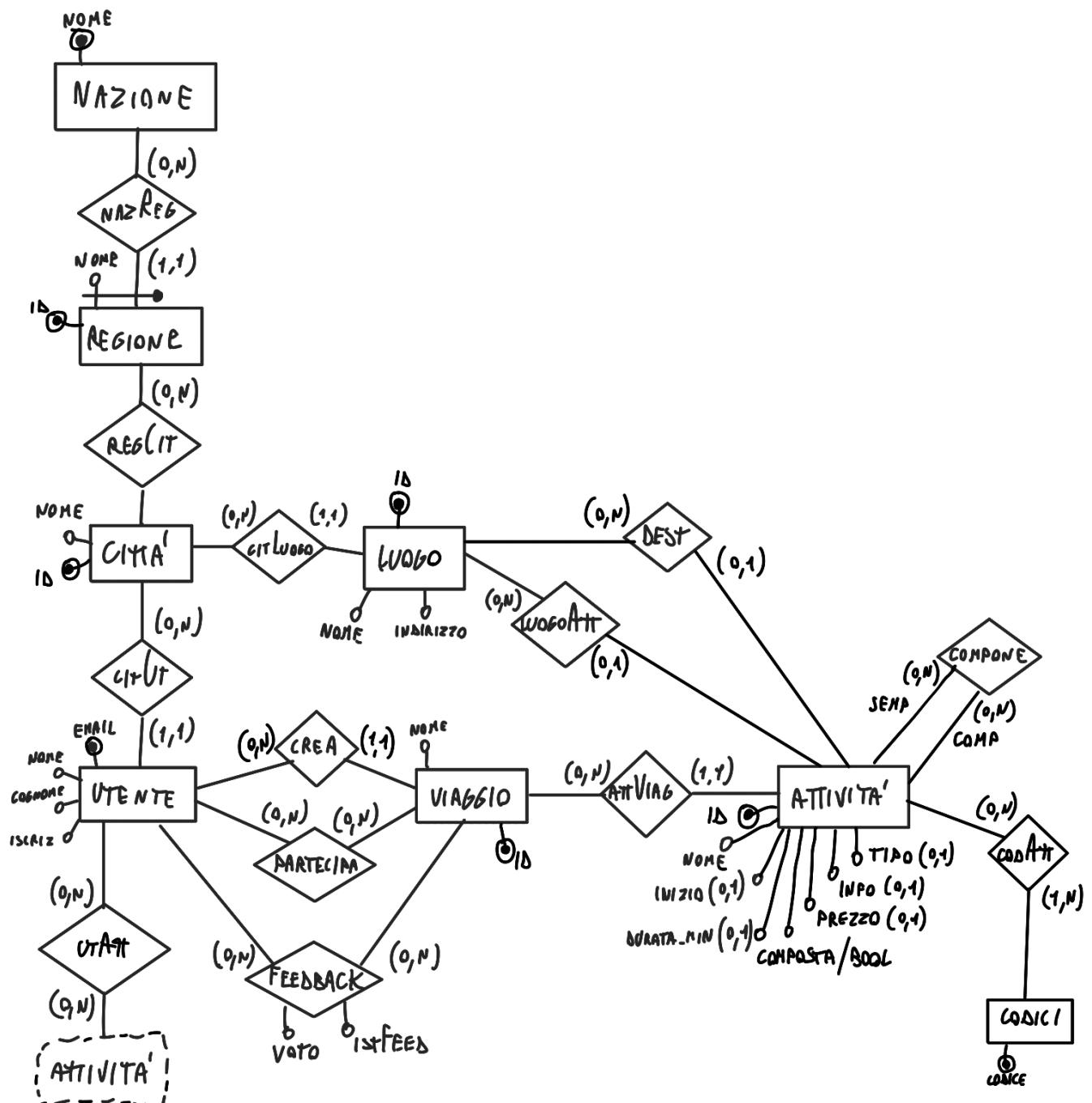
Corrispondenza tra domini concettuali e domini supportati dal DBMS

- CREATE DOMAIN INT_GEZ AS INTEGER CHECK(VALUE >= 0)
- CREATE DOMAIN STR AS VARCHAR(200)
- CREATE DOMAIN EMAIL AS STR CHECK (ISVALID(VALUE))
- CREATE DOMAIN INT_GZ AS INTEGER CHECK (VALUE > 0)
- CREATE DOMAIN Voto AS INTEGER CHECK (VALUE >= 1 AND VALUE <= 5)
- CREATE DOMAIN REAL_GEZ AS REAL
- CREATE TYPE INDIRIZZO AS (

 VIA STR,
 CIVICO INT_GZ,
 CAP INT_GZ
)
- CREATE TYPE DENARO AS (

 VALUTA CHAR(3),
 IMPORTO REAL_GEZ
)
- CREATE TYPE TIPO_ATT AS ENUM('SPOST', 'PERN', 'PRANZO', 'TOUR', 'VISITA')

Diagramma ER ristrutturato



Breve descrizione delle scelte effettuate durante la ristrutturazione

- Accorciamento ATTRIBUTO COMPOSTA è ATTRIBUTO SEMPLICE IN ATTIVITÀ' CON AGGIUNTA DI VINCOLI ESTERNI PER PRESERVARE LA LOGICA E UN ATTRIBUTO BOLEANO PER PRESERVARE LA GENERALIZZAZIONE

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)

x • V. ATTIVITÀ'. SEMPLICE

$$\forall a \text{ ATTIVITÀ'}(a) \rightarrow (\exists p, i, t, l, d \text{ PREZZO}(a, p) \wedge \text{luogoAtt}(a, l) \wedge \text{tiro}(a, t) \wedge \text{inizio}(a, i) \wedge \text{durata-min}(a, d) \leftrightarrow \text{COMPOSTA}(a, \text{FALSE}))$$

T • V. ATTIVITÀ'. COMPOSTA

$$\forall a, a' \text{ ATTIVITÀ'}(a) \wedge \text{ATTIVITÀ'}(a') \wedge \text{COMUNE}(\text{SEMPL}: a', \text{com}: a) \rightarrow \text{COMPOSTA}(a, \text{TRUE}) \wedge \text{COMPOSTA}(a', \text{FALSE})$$

x • V. ATTIVITÀ'. SOSTAMENTO

$$\forall a \text{ ATTIVITÀ'}(a) \wedge \text{tiro}(a, 'SOST') \rightarrow \exists l \text{ DEST}(a, l)$$

T • V. PIANOTAKENTO. UNO AL GIORNO

$$\forall u, p, p', i, i', d, d' \text{ UTENTE}(u) \wedge \text{ATTIVITÀ'}(p) \wedge \text{ATTIVITÀ'}(p') \wedge \text{tiro}(p, 'PERN') \wedge \text{tiro}(p', 'PERN') \wedge p \neq p' \wedge u \notin \text{PARTECIPANTI}(p) \wedge u \in \text{PARTECIPANTI}(p') \wedge \text{inizio}(p, i) \wedge \text{data}(i, d) \wedge \text{inizio}(p', i') \wedge \text{data}(i', d') \rightarrow d \neq d'$$

x • V. LUOGO. SOSTAMENTO

$$\forall a, l, l' \text{ ATTIVITÀ'}(a) \wedge \text{tiro}(a, 'SOST') \wedge \text{luogoAtt}(a, l) \rightarrow \neg \text{DEST}(a, l)$$

Risposta alla Domanda 6 (segue)

- V. ATTRAMAOSTA. VIAGGIO

$\forall a, c, z_a, v \text{ ATTRIVITA}'(a, c) \wedge \text{COMPOSTA}(a, c, \text{TRUE}) \wedge \text{CONDONE}(a, z_a) \wedge \text{ATTRVIAGGIO}(v, a, z_a) \rightarrow \text{ATTRVIAGGIO}(v, a, z_a)$

- V. ATTRAMAOSTA. UTENTE

$\forall a, c, z_a, u \text{ ATTRIVITA}'(a, c) \wedge \text{COMPOSTA}(a, c, \text{TRUE}) \wedge \text{CONDONE}(a, z_a) \wedge \text{UTATTR}(z_a, u) \rightarrow \text{UTATTR}(a, u)$

- V. FEEDBACK. ISA

$\forall u, v \text{ UTENTE}(u) \wedge \text{FEEDBACK}(u, v) \rightarrow \text{PARTECIPA}(u, v)$

- V. ATTRIVITA'. CODICI

$\forall a \text{ ATTRIVITA}'(a) \rightarrow (\exists c \text{ CODATTI}(a, c) \rightarrow \text{COMPOSTA}(a, c, \text{FALSE}))$

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.

<input type="checkbox"/> Relazione ... NAZIONE (nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>NAME</u>	
Domini STR	

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:

<input type="checkbox"/> Relazione ... REGIONE (nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>ID</u> NAME NAZ	
Domini INTEGER STR STR	

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):

SERIAL: ID
CHIAVE: (NAME, NAZ)
FK: (NAZ) REF NAZIONE(NAME)

La relazione accorda le relazioni che implementano le seguenti relationship: **NAZREG**

<input type="checkbox"/> Relazione CITTA' (nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>ID</u> NAME REG	
Domini INTEGER STR 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: (REG) REF REGIONE (ID)
SERIAL: ID

La relazione accorda le relazioni che implementano le seguenti relationship: **REGCITTA'**

<input type="checkbox"/> Relazione ... UTENTE (nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>EMAIL</u> NAME COGNOME ISCRIZ CITTA	
Domini EMAIL STR STR DATE 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 accorda le relazioni che implementano le seguenti relationship: **CITUT**

<input type="checkbox"/> Relazione ... VIAGGIO (nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>ID</u> NAME CREATORE	
Domini INTEGER STR EMAIL	

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):

SERIAL: ID
FK: (CREATORE) REF UTENTE(EMAIL)

La relazione accorda le relazioni che implementano le seguenti relationship: **CREA**

6 Relazione	<u>PARTECIMA</u>	(nome)	Derivante da:	entità	relationship	(cerchiare)
Attributi	<u>UTENTE</u>	<u>VIAGGIO</u>				
Domini	EMAIL	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(EMAIL)

FK: (VIAGGIO) REF VIAGGIO(ID)

La relazione accorda le relazioni che implementano le seguenti relationship:

7 Relazione	<u>FEEDBACK</u>	(nome)	Derivante da:	entità	relationship	(cerchiare)
Attributi	<u>UTENTE</u>	<u>VIAGGIO</u>	<u>VOTO</u>	<u>ISYFEED</u>		
Domini	EMAIL	INTEGER	VOTO	TIMESTAMP		

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, VIAGGIO) REF PARTICIPA(UTENTE, VIAGGIO)

La relazione accorda le relazioni che implementano le seguenti relationship:

8 Relazione	<u>ATTIVITA'</u>	(nome)	Derivante da:	entità	relationship	(cerchiare)
Attributi	<u>ID</u>	<u>NOME</u>	<u>INIZIO</u> *	<u>DURATA_MIN</u> *	<u>COMPSTA</u>	<u>LUOGO</u> * <u>TIPO</u> * <u>PREZZO</u> *
Domini	INTEGER	STR	TIMESTAMP	INT-62	BOOLEAN	INTEGER TIPO-ATT DENARO
					<u>VIAGGIO</u> * <u>INFO</u> * <u>DEST</u> *	
					INTEGER STR	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) ENN: DEST ≠ NULL \Leftrightarrow INFO = 'POST' SERIAL: ID

FK: (VIAGGIO) REF LUOGO(ID) ENN: DEST ≠ NULL \rightarrow DEST ≠ LUOGO

FK: (DEST) REF LUOGO(ID) ENN: COMPSTA = FALSE $\Leftrightarrow TIPO ≠ NULL \wedge LUOGO ≠ NULL \wedge INIZIO ≠ NULL \wedge DURATA_MIN ≠ NULL$

La relazione accorda le relazioni che implementano le seguenti relationship: ...ATTIVAG, LUOGOAtt, DEST....

9 Relazione	<u>CODICE</u>	(nome)	Derivante da:	entità	relationship	(cerchiare)
Attributi	<u>CODICE</u>					
Domini	STR					

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):

INC: (CODICE) ⊑ costr(COD)

La relazione accorda le relazioni che implementano le seguenti relationship:

10 Relazione	<u>CODAtt</u>	(nome)	Derivante da:	entità	relationship	(cerchiare)
Attributi	<u>ATT</u>	<u>COD</u>				
Domini	INTEGER	STR				

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: (ATT) REF ATTIVITA'(ID)

FK: (COD) REF CODICE(CODICE)

La relazione accorda le relazioni che implementano le seguenti relationship:

11 Relazione ... <u>LUGO</u>(nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>ID</u> <u>NOME</u> <u>CITTA'</u> <u>(NRIZZO)</u>	
Domini <u>INTEGER</u> <u>STR</u> <u>INTEGER</u> <u>INNIZIAZIO</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: (CITTA') REF CITTA'(ID)

SERIAL:ID

La relazione accorda le relazioni che implementano le seguenti relationship: CITLUGO

12 Relazione ... <u>COMPONE</u>(nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>COMPONTE</u> <u>SEMPLICE</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):

fk: (COMPONTE) REF ATTIVITA'(ID)

fk: (SEMPLICE) REF ATTIVITA'(ID)

La relazione accorda le relazioni che implementano le seguenti relationship:

13 Relazione ... <u>UTAtt</u>(nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>UTENTE</u> <u>ATTIVITA'</u>	
Domini <u>EMAIL</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):

fk: (UTENTE) REF UTENTE(EMAIL)

fk: (ATTIVITA') REF ATTIVITA'(ID)

La relazione accorda 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 ennupla, di dominio):

La relazione accorda 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 ennupla, di dominio):

La relazione accorda 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 accorda 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 accorda 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 accorda 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 accorda 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 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 ennupple); (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.

- REVOKE UPDATE ON ATTIVITA' FROM USER

- T. PERNOTTAMENTO.UNO AL GIORNO

 - INSERIMENTO IN ATTIVITA'

 - POST-OPERAZIONE

$$\text{ISERROR} = \left(\begin{array}{l} \text{EXISTS} \left(\begin{array}{l} \text{SELECT } U.\text{EMAIL} \text{ FROM } \text{ATTIVITA}' a, U\text{TENTE } U \\ \text{WHERE } \text{NEW.TIPO} = 'PERN' \text{ AND } a.\text{TIPO} = 'PERN' \text{ AND } \\ U.\text{EMAIL} \text{ IN DB.PARTECIPANTI}(\text{NEW.ID}) \text{ AND } \\ U.\text{EMAIL} \text{ IN DB.PARTECIPANTI}(a.\text{ID}) \text{ AND } \\ \text{AND EXTRACT(DATE FROM NEW.INIZIO)} = \\ \text{EXTRACT(DATE FROM a.INIZIO)} \end{array} \right) \end{array} \right)$$

IF ISERROR

REVERT

GENERA ERRORE

- REVOKE UPDATE ON COMPOSIZIONE FROM USER

- T. ATTIVITA'. COMPOSTA

 - INSERIMENTO IN COMPOSIZIONE

 - PRE-OPERAZIONE

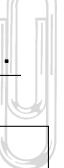
$$\text{ISVALID} = \left(\begin{array}{l} \text{EXISTS} \left(\begin{array}{l} \text{SELECT } * \text{ FROM } \text{ATTIVITA}' a, \text{ATTIVITA}' a' \\ \text{WHERE } \text{NEW.COMPOSTA} = a.\text{ID} \text{ AND } \text{NEW.SEMPLICE} = a'.\text{ID} \\ \text{AND } a.\text{COMPOSTA} = \text{TRUE} \text{ AND } a'.\text{COMPOSTA} = \text{FALSE} \end{array} \right) \end{array} \right)$$

IF ISVALID

COMMIT

ELSE

GENERA ERRORE



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

- DB. CALCOLA_PUNTEGGIO (*a: EMAIL*): INT-GEZ

$$Q = \left(\text{SELECT AVG}(V.VOTO) AS M, \right. \\ \left. \text{PL00A}\left(0.1 * \text{SUM}(\text{CASE WHEN } f.VOTO > 4 \text{ THEN } 1 \text{ ELSE } 0 \text{ END})\right) \text{ AS } \gamma \right. \\ \text{FROM VIAGGIO } V, \text{FEEDBACK } f \\ \text{WHERE } V.CREATOR = :u \text{ AND } f.VIAGGIO = V.ID \\ \text{IF } Q.M \leq 3 \\ \text{RETURN } 0 \\ \text{ELSE} \\ \text{RETURN } Q.\gamma$$

- DB. INIZIO_VIAGGIO (*v: INTEGER*): TIMESTAMP

$$Q = \left(\text{SELECT MIN}(a.INIZIO) AS i \text{ FROM ATTIVITA' } 2 \right. \\ \left. \text{WHERE } a.COMPOSITA = \text{FALSE AND } a.VIAGGIO} = :v \right) \\ \text{RETURN } Q.i$$

- DB. FINE_VIAGGIO (*v: INTEGER*): TIMESTAMP

$$Q = \left(\text{SELECT MIN}(a.INIZIO + a.DURATA_MIN * '1 MINUTE'::INTERVAL) AS f \text{ FROM ATTIVITA' } 2 \right. \\ \left. \text{WHERE } a.COMPOSITA = \text{FALSE AND } a.VIAGGIO} = :v \right) \\ \text{RETURN } Q.f$$

Risposta alla Domanda 8 (segue)

RICERCA VIAGGI PARAMETRI ($R: \text{INSIENE}(\langle z: \text{INTEGER} \rangle), i: \text{TIMESTAMP}, f: \text{TIMESTAMP},$
 $b_{\min}: \text{DENARO}, b_{\max}: \text{DENARO}, p: \text{INT-GEZ} : \text{INSIENE}(\langle v: \text{INTEGER} \rangle)$)

IF $(:f > :i) \text{ OR } (:b_{\max} \geq :b_{\min})$

GENERA ERRORE

TEMP VIAGGI (V: INTEGER)

FK: (v) REF VIAGGIO (id)

INSERT INTO VIAGGI / VIAGGI(V)

```
(SELECT V.ID
  FROM VIAGGI V, ATTIVITA a, LUOGO l, CITTÀ c
 WHERE :i <= DB.INIZIOVIAGGIO(V.ID) AND DB.FINEVIAGGIO(V.ID) <= :f
   AND a.VIAGGIO = V.ID AND (a.LUOGO = l.ID OR a.DEST = l.ID)
   AND l.CITTÀ = c.ID AND c.REG IN (:R))
```

$Q = \left(\begin{array}{l} \text{SELECT } V \\ \text{FROM (SELECT } V.V \text{ AS } V, \\ \text{SUM(a.PREZZO)} \text{ AS } b, \\ \text{DB.CALCOLA PUNTEGGIO(V.CREATORE)} \text{ AS } p' \\ \text{FROM VIAGGI/REGIONI } V, \text{ATTIVITA' } a \\ \text{GROUP BY } V.V \\ \text{HAVING } :MIN \leq b \text{ AND } b \leq :MAX \\ \text{AND } p' \geq :P \end{array} \right)$

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.]

NUMVIAGGIULTIMOANNO ($c: \text{INTEGER}$): INSIEME ($\langle m: \text{INT_62}, n: \text{INT_62} \rangle$)

TMPMESI ($m: \text{INT_62}$)

WITH RECURSIVE GENMESI (n) AS (

VALUES (1)

UNION ALL

SELECT $n+1$ FROM GENMESI

WHERE $n < 12$)

INSERT INTO TMPMESI (m)

SELECT n FROM GENMESI

TMPVIAGGI ($v: \text{INTEGER}, i: \text{TIMESTAMP}, f: \text{TIMESTAMP}$)
 FK: (v) REF VIAGGIO (id)

INSERT INTO TMPVIAGGI (v, i, f)

SELECT $v.id, db.inizioviaggio(v.id)$ AS $i, db.fineviaggio(v.id)$ AS f
 FROM VIAGGIO v

GROUP BY $v.id$

HAVING (CURRENT_TIMESTAMP - '1 YEAR'::INTERVAL) $<= i$

AND $f <= \text{CURRENT_TIMESTAMP}$

$Q = \left(\begin{array}{l} \text{SELECT tm.m, COUNT(tv.v)} \\ \text{FROM TMPMESI tm, TMPVIAGGIO tv, ATTIVITA a, LUOGO l} \\ \text{WHERE EXTRACT(MONTH FROM tv.i)} = tm.m \\ \text{AND a.viaggio} = tv.v \text{ AND (a.luogo} = l.id} \\ \text{OR a.dest} = l.id) \wedge l.citta' = :c \\ \text{GROUP BY tm.m} \end{array} \right)$

RETURN Q

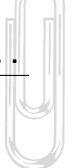
[Spazio per minute. Questa pagina non sarà valutata a meno che non sia puntata da pagine precedenti.]

$\text{NUMVIAGGI}(\text{REGIONI NAZIONE}(\text{n:STR}, \text{i:TIMESTAMP}, \text{f:TIMESTAMP}) : (\text{NSIENE}(\text{r-1:INTEGER}, \text{n':INT-BEZ}))$
 IF $f < i$:
 GENERA ERRORE

$Q = (\text{SELECT r.ID, COUNT(v.ID)}$
 $\text{FROM REGIONE r, VIAGGIO v, ATTIVITA a, LUOGO l, CITTÀ c}$
 $\text{WHERE r.NAZ = :n AND :i <= DB.INIZIOVIAGGIO(v.ID)}$
 $\text{AND DB.FINEVIAGGIO(v.ID)} \leq :f \text{ AND a.VIAGGIO = v.ID}$
 $\text{AND (a.LUOGO = l.ID V a.DEST = l.ID)}$
 $\text{AND l.CITTÀ = c.ID AND c.REG = r.ID}$
 $\text{GROUP BY r.ID})$

RETURN Q

[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.]

