



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

Esame BD2.Esame.Risposte – Modulo risposte prova scritta (diagramma delle classi UML)

Dati dello studente e dell'esame

Cognome e nome: Matricola:

Data:

Corso di laurea e canale di appartenenza:

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

Firma di un membro della Commissione per
avvenuta identificazione:

.....

Rinuncia alla prova

Desidero rinunciare a questa prova d'esame. Firma:



Istruzioni e regole d'esame

Prima dell'esame

- Stampare questo modulo, preferibilmente fronte-retro, e rilegarlo con un fermaglio rimovibile, come quello disegnato in alto
- Compilare il frontespizio con i propri dati, come richiesto
- Scrivere la propria matricola nello spazio apposito nella parte alta di tutte le pagine

Durante l'esame

- La prova è dimensionata per essere svolta in circa 3 ore. Tuttavia, data la sua natura fortemente progettuale, la Commissione offre agli studenti la più ampia disponibilità di tempo, al fine ovviare ad eventuali (e limitati) errori di analisi/progettazione rilevati più a valle del ciclo di vita.

Il tempo massimo per la consegna è quindi rilassato a 5 ore (il massimo tempo compatibile con le disponibilità di aule).

- Scrivere le risposte negli spazi predisposti sotto le relative domande. Le ultime pagine sono vuote e possono essere usate come minute oppure, se puntate opportunamente, per contenere risposte in caso gli spazi appositi dovessero risultare insufficienti.
- Non è possibile usare alcun tipo di materiale didattico.
- In caso di necessità di ulteriori fogli (in proprio possesso), chiedere preventivamente alla Commissione una nuova procedura di controllo.
- La Commissione può rispondere solo a brevi domande inerenti al testo dei quesiti.
- Tra la seconda e la quarta ora d'esame, gli studenti possono effettuare **brevi pause** (uno studente alla volta) seguendo la seguente procedura:
 1. Alla lavagna è riportata una coda denominata 'Coda prenotazioni pause'. Sia n (un intero) l'elemento in fondo alla coda (si assuma $n = 0$ in caso di coda vuota).
 2. Recarsi alla lavagna ed aggiungere l'intero $n + 1$ come proprio contrassegno in fondo alla coda, seguito da una stringa a propria scelta (ad es., le proprie iniziali).
 3. Se il proprio contrassegno non è l'elemento affiorante della coda, tornare al lavoro in attesa che lo diventi.
 4. Consegnare tutti i fogli di lavoro e il testo d'esame alla Commissione ed uscire.
 5. Al rientro, cancellare il proprio contrassegno dalla coda di modo da permettere al successivo studente prenotato di uscire, e riprendere i fogli prima consegnati.

Al momento della consegna

- Ordinare tutti i fogli che si vuole far valutare e rilegarli con un fermaglio rimovibile. Non includere fogli che la Commissione non deve valutare (ad es., requisiti, minute), ma includere ovviamente il frontespizio.
- Consegnare i fogli ordinati **nelle mani** di un membro della Commissione. **Non lasciare l'aula senza la conferma, da parte della Commissione, del buon esito delle operazioni di consegna.**

In caso di rinuncia

- È possibile rinunciare alla consegna a partire dalla seconda ora d'esame. In caso di rinuncia, consegnare nelle mani della Commissione solo il frontespizio, dopo aver compilato e firmato la sezione dedicata.

Sommario delle domande

Si richiede di progettare l'applicazione descritta dalla specifica dei requisiti effettuando le fasi di Analisi concettuale dei requisiti e di Progettazione logica della base dati e delle funzionalità, utilizzando la metodologia vista nel corso.

In particolare (vengono indicati i tempi suggeriti per i diversi passi chiave):

Parte 1: Analisi concettuale dei requisiti Effettuare la fase di Analisi concettuale dei requisiti producendo lo schema concettuale per l'applicazione, che includa:

- Analisi dei dati (45 minuti; 75 minuti al massimo):
 - un diagramma UML concettuale delle classi (*)
 - (parte del)le specifiche formali delle classi e delle associazioni
 - le specifiche dei tipi di dato
 - la specifica formale dei vincoli esterni (*)
- Analisi delle funzionalità:
 - un diagramma UML degli use-case (5 minuti; 10 minuti al massimo)
 - la segnatura di tutte le operazioni di use-case (10 minuti)
 - (parti del)le specifiche formali degli use-case. (30 minuti; 60 minuti al massimo)

Si richiede *esplicitamente* di modellare le specifiche formali delle operazioni di classe e/o use-case necessarie a modellare i requisiti contrassegnati dalla barra laterale (come quella qui a sinistra), *incluso* tutte le eventuali operazioni ausiliarie, usando l'estensione della logica del primo ordine studiata nel corso. (*)

Parte 2: Progettazione della base dati e delle funzionalità Effettuare la progettazione della base dati e delle funzionalità a partire dallo schema concettuale prodotto nella Parte 1, ed in particolare eseguire i seguenti passi:

- Progettazione della base dati relazionale con vincoli:
 - Ristrutturazione del diagramma UML concettuale delle classi e delle specifiche (20 minuti; 30 minuti al massimo):
 - * scelta del DBMS da utilizzare
 - * progettazione della corrispondenza tra i tipi di dato concettuali ed opportuni domini SQL (domini base o utente, oppure realizzati mediante relazioni aggiuntive) supportati dal DBMS scelto
 - * ristrutturazione del diagramma UML concettuale delle classi e delle specifiche dei vincoli esterni.
 - Produzione dello schema relazionale della base dati e dei relativi vincoli (*) (30 minuti; 60 minuti al massimo)
- Progettazione delle funzionalità (30 minuti; 45 minuti al massimo):
 - definizione della specifica realizzativa delle operazioni necessarie a modellare i requisiti contrassegnati dalla barra laterale, in modo conforme alla loro specifica concettuale prodotta nella fase di Analisi, in termini di algoritmi in pseudo-codice e comandi SQL immersi. (*)

Le pagine seguenti contengono le domande specifiche a cui è richiesto rispondere, ulteriori delucidazioni per ogni singolo punto, e spazi per le risposte.

Le pagine da 31 in poi possono essere utilizzate per scrivere minute che non verranno valutate.

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



Questa pagina è stata intenzionalmente lasciata vuota

1 Analisi concettuale

Domanda 1 (10 minuti) Raffinare la specifica dei requisiti eliminando inconsistenze, omissioni e ridondanze e producendo un elenco numerato di requisiti il meno ambiguo possibile. (La risposta a questa domanda non sarà valutata, ma si consiglia di svolgere accuratamente questo passo, in quanto può facilitare di molto le attività di progetto.)

Risposta

Utente

- Nome
- Cognome
- DATA NASCITA
- INDIREZIA
- EMAIL
- CREA

Città

- Nome
- Nazionalità

Nazionalità

- nome

Tributaria

- Nome
- PESO
- DATI BIODATI
- STIPULABILE DDA
- STIPULABILE A
- MAX UGOLI

CLIENTE FISICO

CLIENTE IMPRESA

ACQUAIO

- Nome
- Cognome
- DATA NASCITA
- DIVA [ca.1]

- DIVA
- RAISON SOCIAL

- TUBO

Accesso

- ENTRATA
- SULCA
- RESTAZIONE

Postazione

- IB:

Servizio

- nome
- DESCRIZIONE
- PREZZO UNITARIO

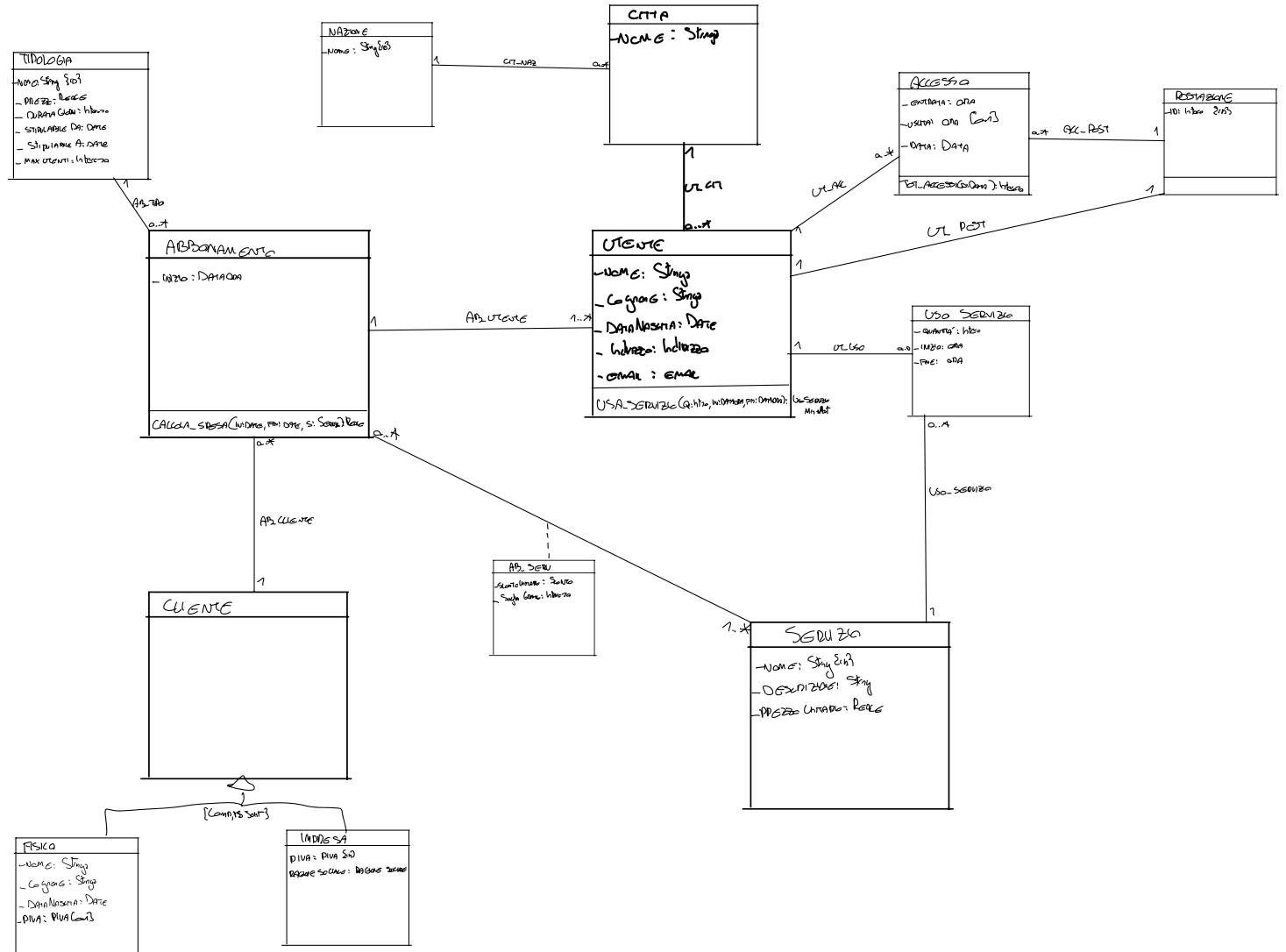
Risposta alla Domanda 1 (segue)

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

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

Diagramma UML concettuale delle classi

Produrre un diagramma UML concettuale delle classi per l'applicazione in termini di classi, associazioni, attributi, generalizzazioni, operazioni di classe.



[continua alla pagina seguente]

Risposta alla Domanda 2 (segue)

Specifiche delle classi o associazioni Per ogni classe o associazione del diagramma **con** operazioni o vincoli:

- Definire la specifica formale di eventuali operazioni necessarie a modellare i requisiti contrassegnati dalla barra laterale, ed eventuali vincoli esterni. Usare la logica del primo ordine estesa con teoria degli insiemi e semantica di mondo reale vista nel corso, usando il seguente alfabeto:
 - Un simbolo di predicato $C/1$ per ogni classe C .
Semantica di $C(x)$: x è una istanza di C .
 - Un simbolo di predicato $T/1$ per ogni tipo di dato T .
Semantica di $T(x)$: x è un valore di T .
 - Un simbolo di predicato $\text{assoc}/2$ per ogni associazione binaria assoc.
Semantica di $\text{assoc}(c_1, c_2)$: (c_1, c_2) è una istanza di assoc.
 - Un simbolo di predicato $\text{attr}/2$ per ogni attributo attr di entità
Semantica di $\text{attr}(c, v)$: uno dei valori dell'attributo attr dell'istanza c è v .
 - Un simbolo di predicato $\text{attr}/3$ per ogni attributo attr di associazione binaria.
Semantica di $\text{attr}(c_1, c_2, v)$: uno dei valori dell'attr. attr del link (c_1, c_2) è v .
 - Un simbolo di predicato $\text{op}/(n+2)$ per ogni operazione di classe ad n argomenti.
Semantica di $\text{op}(c, \text{arg}_1, \dots, \text{arg}_n, v)$: uno dei valori di ritorno di op, quando invocata sull'istanza c e con argomenti $\text{arg}_1, \dots, \text{arg}_n$ è v .
 - Il simbolo di $=/2$ (la cui interpretazione è la relazione che lega ogni elemento del dominio di interpretazione solo con se stesso) e opportuni simboli di predicato e di funzione, soggetti a semantica di modo reale, per relazioni e funzioni standard tra elementi dei tipi di dato, tra cui $\text{adesso}/0$, interpretato come il valore del dominio DataOra che rappresenta l'istante corrente.

Risposta

<p>1 Tipo: <u>Classe</u> Associazione (cerchiare)</p> <p>Nome: ...TIPOLOGIA.....</p> <p>Operazioni, vincoli:</p> <p><u>[U. TIPOLOGIA. CONTINUITÀ]</u></p> <p>$\forall T, SD, SA$</p> <p>$\{ \text{Tipologia}(T) \wedge \text{Stipubblica}(T, SD) \wedge \text{Stipubblica}(T, SA) \} \rightarrow SD = SA$</p> <p><u>[U. TIPOLOGIA. MAX UTENZE]</u> T</p> <p>$\forall A, T, SD, SA$</p> <p>$\{ \text{Tipologia}(A) \wedge \text{Abbonamento}(A, T) \wedge \text{Abbonamento}(A, SD) \wedge \text{Abbonamento}(A, SA) \} \rightarrow \text{MaxUtenti}(A, T) = SD = SA$</p> <p>$\forall T, m$</p> <p>$\{ \text{Tipologia}(T) \wedge \text{MaxUtenti}(T, m) \} \rightarrow \{ A \} \subseteq M$</p>	<p>2 Tipo: <u>Classe</u> Associazione (cerchiare)</p> <p>Nome: ...ABBONAMENTO.....</p> <p>Operazioni, vincoli:</p> <p><u>[U. ABBONAMENTO. NUOVA]</u> T</p> <p>$\forall A, T, SD, SA$</p> <p>$\{ \text{Abbonamento}(A) \wedge \text{Tipologia}(A) \wedge \text{Abbonamento}(A, T) \wedge \text{Abbonamento}(A, SD) \wedge \text{Abbonamento}(A, SA) \} \rightarrow SD = SA$</p>
--	---

<p><input type="checkbox"/> Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ACCESSO</p> <p>Operazioni, vincoli:</p> <p>[C. ACCESSO. CONTINUA] T</p> <p>$\forall A, \exists v$</p> <p>$(\text{Accesso}(A) \wedge \text{ENTRATA}(A, E) \wedge \text{CER}(A, E)) \rightarrow \text{ELO}$</p> <p>[C. ACCESSO. NASCITA UT]</p> <p>$\forall A, \exists v, \exists u$</p> <p>$(\text{Accesso}(A) \wedge \text{ENTRATA}(A, E) \wedge \text{UT_ACC}(u, A) \wedge \text{UT_REC}(u, A)) \rightarrow \text{IDLE}$</p> <p>$\wedge \text{DATA_NASCITA}(u, v) \rightarrow \text{IDLE}$</p>	<p><input type="checkbox"/> Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ACCESSO</p> <p>Operazioni, vincoli:</p> <p>[C. ACCESSO. DESTINAZIONE] T</p> <p>$\forall A, \exists v, \exists p$</p> <p>$(\text{Accesso}(A) \wedge \text{UT_ACC}(v, A) \wedge \text{POSTAZIONE}(p) \wedge \text{UT_ACC}(v, p) \wedge \text{ACCEP}(A, p)) \rightarrow \text{UT_DEST}(v, p)$</p>
<p><input type="checkbox"/> Tipo: Classe Associazione (cerchiare)</p> <p>Nome: USO SERVIZIO</p> <p>Operazioni, vincoli:</p> <p>[C. USO SERVIZIO. CONTINUA]</p> <p>$\forall U, \exists F$</p> <p>$(\text{Uso_SERVIZIO}(U) \wedge (\text{WIZ}(U, I) \wedge \text{FINE}(U, F))) \rightarrow \text{ILF}$</p>	<p><input type="checkbox"/> Tipo: Classe Associazione (cerchiare)</p> <p>Nome: UTENTE</p> <p>Operazioni, vincoli:</p> <p>[C. UTENTE. NASCITA]</p> <p>$\forall G, \exists N, \exists A$</p> <p>$(\text{UTENTE}(G) \wedge \text{DATA_NASCITA}(N, G) \wedge \text{Albano}(A, G) \wedge \text{WIZ}(A, G)) \rightarrow \text{NCL}$</p>
<p><input type="checkbox"/> Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ABBONAMENTO</p> <p>Operazioni, vincoli:</p> <p>[C. ABBONAMENTO. ACCESSI] T</p> <p>$\forall A, \exists U, \exists F$</p> <p>$(\text{ABBONAMENTO}(A) \wedge \text{INIT}(A, I) \wedge \text{UT_REC}(U, A) \wedge \text{AB_UT}(A, U) \wedge \text{ACCESS}(A) \wedge \text{UT_ACC}(U, F) \wedge \text{ENTRATA}(A, E)) \rightarrow \text{ILS}$</p>	<p><input type="checkbox"/> Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ABBONAMENTO</p> <p>Operazioni, vincoli:</p> <p>[C. ABBONAMENTO. FINE]</p> <p>$\forall A, \exists I, \exists B, \exists U, \exists G, \exists E$</p> <p>$(\text{ABBONAMENTO}(A) \wedge \text{INIT}(A, I) \wedge \text{TIPO_GAR}(G) \wedge \text{AB_TP}(A, I) \wedge \text{DURATA_GAR}(G, D) \wedge \text{UT_REC}(U, A) \wedge \text{AB_UT}(A, U) \wedge \text{ACCESS}(A) \wedge \text{UT_ACC}(U, F) \wedge \text{ENTRATA}(A, E)) \rightarrow \text{E} \leq (+D)$</p>

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

TIPI DI DATO

- DATA: $\{ \text{data} \}$
- INDIRIZZO_CAPP: $\{ \text{data}: \text{String}, \text{vincolo: Intero} \}$
- EMAIL = STONKA SECONDO REGEX
- PIVA = STONKA SECONDO REGEX
- RADICE_SOCIALE = STONKA SECONDO REGEX
- SCONTO = REALE con OLTRE VALORE $\neq 1$

ULTERIORI VINCOLI

V. CLIENTE, NA_SVRA

$\{ \text{V. CLIENTE, NA_SVRA} \}$

$\{ \text{V. CLIENTE, NA_SVRA} \} \cap \text{ABONAMENTO} \cap \text{ABONAMENTO}(\text{C}_1) \cap \text{ABONAMENTO}(\text{C}_2) \cap \text{ABONAMENTO}(\text{C}_3) \rightarrow \text{VCL}$

$\{ \text{FISICO}(\text{C}) \cap \text{DATA_NAZIONA}(\text{C}, \text{N}) \cap \text{ABONAMENTO} \}$ AN VINCOLI

OPERAZIONI DI CLASSE

CLASSE ABONAMENTO

$\text{CALCOLA_SVRA}(\text{In: DataInizio, Fin: DataFin, S: Scenari}): \text{Data} =$

OPRE: INFERIRE $\exists S \text{ Scenari}(S)$

O Post: $M_n = M_{n+1}$

$S_n = \sum_{i=1}^n \left\{ \begin{array}{l} \text{UT}_i \\ \text{PU}_i \end{array} \right\} \mid \begin{array}{l} \text{UT}_i \cap \text{AB_UT}(\text{Th}_i, \text{UT}_i) \cap \text{US_SERVIZIO}(\text{UT}_i) \cap \text{UT_UT}(\text{UT}_i, \text{UT}_{i+1}) \cap \text{US_SERVIZIO}(\text{UT}_i, \text{S}) \\ \text{PU}_i \cap \text{PREZZO UNITARIO}(\text{S}, \text{PU}_i) \end{array} \} \cap \text{UT}_i \cap \text{PU}_i \cap \text{SERVIZIO}(\text{UT}_i, \text{S}) \cap \text{SCENARI}(\text{UT}_i, \text{S}, \text{SC})$

$S_n = \text{UT}_1 + \dots + \text{UT}_n = \left(\sum_{i=1}^n \text{UT}_i \right) - \text{SG}$

$$R = ((P_G \cdot (1 - P_U)) \cdot \text{TOT_UTILIZZO})$$

$$R_{\text{RESULT}} = R$$

CLASSE Accesso

$\text{TOT_ACCESSI}(\text{D: DATA}): \text{Intero}$

OPRE: NO SUMMA

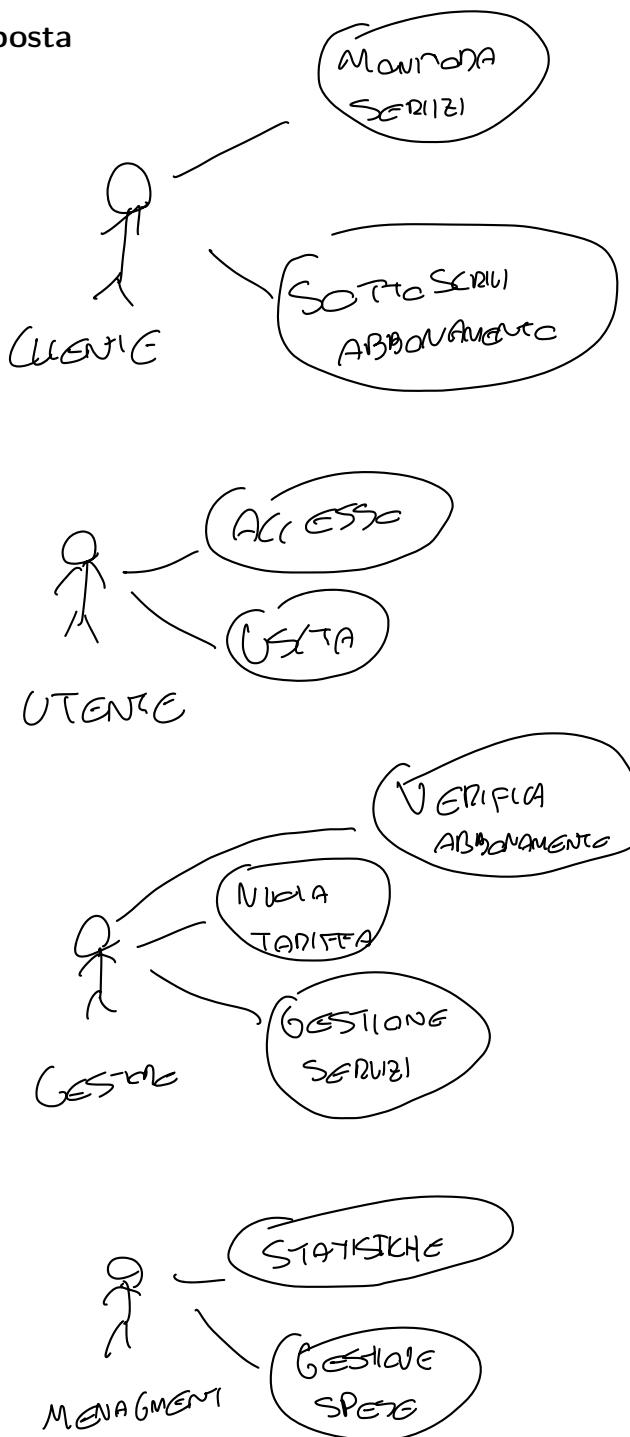
O Post: $M_n = M_{n+1}$

$R_{\text{RESULT}} = \left| \left\{ A \mid \text{ACCESSO}(\text{A}) \cap \text{DATA}(\text{A}, \text{DATA}) \cap \text{DATA} = D \right\} \right|$

[continua alla pagina seguente]

Risposta alla Domanda 2 (segue)

Domanda 3 (5 minuti; 10 minuti al massimo) Proseguire la fase di Analisi Concettuale dei requisiti, producendo un diagramma UML degli use-case che definisca ad alto livello tutte le funzionalità richieste al sistema.

Risposta



Questa pagina è stata intenzionalmente lasciata vuota

Domanda 4 (10 minuti) Proseguire la fase di Analisi Concettuale dei requisiti definendo la **segnatura** delle operazioni in ogni use-case.

Risposta

USE CASE MONITORA SERVIZI

MONITORA (): [SERVIZIO] [o..1]

USE CASE SOTTO SERVIZI ABBONAMENTO

SOTTO-SERVIZIO [T: TIPOLOGIA, U: UTENTE] [] : ABBONAMENTO

USE CASE ACCESSO

ACCESSO (): ACCESSO

USE CASE USCITA

USCITA (): USCITA

USE CASE NUOVA TARIFFE

NUOVA-TARIFFA (N: STRING, P: DECIMAL, D6: Integer, SORNAME, Surname, M: Integer) : TARIFFE

USE CASE VERIFICA ABBONAMENTO

VERIFICA (U: UTENTE) : Bool

USE CASE GESTIONE SERVIZI

NUOVO-SERVIZIO (N: STRING, D: STRING, P: DECIMAL) : SERVIZIO

ELIMINA-SERVIZIO (S: SERVIZIO)

USE CASE STATI-STICHE

STATI-STICHE ([IN: TIME, FIN: TIME] [o..1]) : ([IN: TIME, FIN: TIME], N) [o..1]

USE CASE GESTIONE SPESE

SPESE (I: Imposta, IN: DATAODA, FM: DATAODA) : REIAS



Questa pagina è stata intenzionalmente lasciata vuota

Domanda 5 (30 minuti; 60 minuti al massimo) Proseguire la fase di Analisi Concettuale dei requisiti producendo le specifiche concettuali per le operazioni di use-case, **limitandosi** a quelle necessarie a modellare i requisiti contrassegnati dalla barra laterale (come quella qui a sinistra), ed includendo eventuali operazioni ausiliarie. In particolare, per ogni operazione, definire segnatura, precondizioni e postcondizioni utilizzando il linguaggio della logica del primo ordine. Si assuma lo stesso vocabolario definito alla Domanda 2.

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

Risposta

SPSC (I: INGRESSO, IN: DATA_IN, FIN: DATA_OUT): REQS

O POST: IN & FIN

O POST = M_{in} = M_{out}

$$\text{SIA } TOT_A = \left\{ \begin{array}{l} A \\ | \quad \text{ABONAMENTO}(A) \wedge \text{AB_CUCINA}(A, 1) \end{array} \right\}$$

$$\text{SIA } TOT_S = \left\{ \begin{array}{l} S \\ | \quad \text{SERVIZIO}(S) \wedge \text{AB_SERV}(A, S) \wedge A \in TOT_A \end{array} \right\}$$

$$POST = \sum_{SETS} \sum_{AGA} \text{CALCOL_SPESA}(A, IN, FIN, S)$$

USE CASE STATISTICHE

STATISTICHE ([I: IN: TIME, FIN: TIME] [G, M]): ([IN: TIME, FIN: TIME], N) [L�]

O PIRE: NESSUNA

O POST:

$$\text{SIA } ADDESSO(A) \cap \left(\exists m \text{ MESE}(m, A) \wedge \text{DATA}(m) \wedge \text{DATA}(DP) \wedge \right. \\ \left. \text{MESE}(DC, m) \wedge \text{MESE}(DF, m) \wedge \text{CUCINA}(DC, 1) \wedge \text{CUCINA}(DF, 3) \right)$$

$$R = \left\{ \begin{array}{l} (I, G, N) \\ | \quad \text{CUCINA}(G) \wedge I \in T \\ | \quad \text{GLOBALE}(G) \wedge \left(\begin{array}{l} \text{GLOBALE}(G) \wedge \text{USCITA}(G, 0) \wedge \text{IN}(I, INIZIO) \wedge \text{FIN}(I, FINE) \\ \text{ACCESO}(G) \wedge \text{ENTRATA}(G, 0) \wedge \text{USCITA}(G, 0) \wedge \text{IN}(I, INIZIO) \wedge \text{FIN}(I, FINE) \end{array} \right) \end{array} \right\} \\ M = \frac{\left\{ \begin{array}{l} A \\ | \quad \text{ACCESO}(A) \wedge \text{ENTRATA}(A, 0) \wedge \text{USCITA}(A, 0) \wedge \text{IN}(I, INIZIO) \wedge \text{FIN}(I, FINE) \end{array} \right\}}{\text{TOT-ACCESSI}(G)} \end{matrix}$$

Risposta alla Domanda 5 (segue)

2 Progettazione della base dati e delle funzionalità

Domanda 6 (20 minuti; 30 minuti al massimo) Iniziare la fase di progettazione logica della base di dati decidendo il DBMS da utilizzare e ristrutturando lo schema UML delle classi concettuale, il dizionario dei dati e i vincoli esterni. In particolare:

- progettare una corrispondenza tra i tipi di dato concettuali ed opportuni domini SQL (domini base o utente, oppure realizzati mediante relazioni aggiuntive) supportati dal DBMS scelto
- eliminare attributi multivale o composti
- eliminare relazioni is-a e generalizzazioni
- definire un identificatore primario per ogni classe
- ristrutturare i vincoli esterni per renderli consistenti con la struttura del nuovo diagramma.

Descrivere brevemente le principali scelte effettuate.

DBMS da utilizzare PostgreSQL

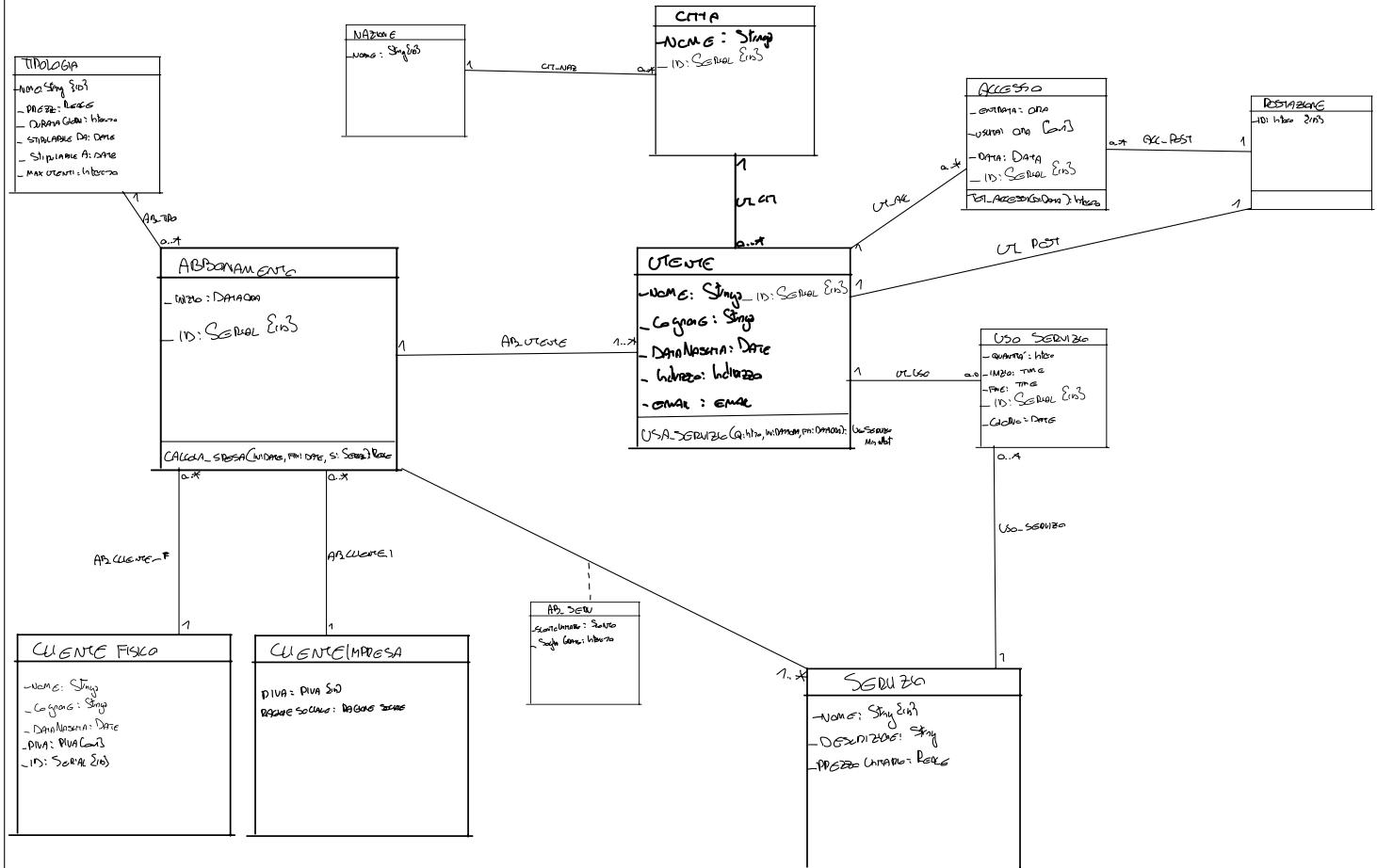
Corrispondenza tra tipi di dato concettuali e domini supportati dal DBMS

```

CREATE DOMAIN STRINGA AS VARCHAR(255) CHECK (VALUE NOT NULL)
CREATE DOMAIN Interzo AS Integer CHECK (VALUE NOT NULL AND VALUE > 0)
CREATE DOMAIN Interzo2 AS Integer CHECK (VALUE NOT NULL AND VALUE > 0)
CREATE DOMAIN Orta AS Integer CHECK (VALUE <= 24)
CREATE DOMAIN Email AS String CHECK (VALUE ~ '^.+\@[a-zA-Z]+\.[a-zA-Z]{2,}$')
CREATE DOMAINPIVA AS STRING CHECK (VALUE ~ '^\d{2}\.\d{2}\.\d{4}$')
CREATE DOMAIN SOCIALE AS String CHECK (VALUE ~ '^\d{11}$')
CREATE DOMAIN DECIMALE AS Real CHECK (VALUE NOT NULL AND VALUE <= 1)
CREATE TYPE Interzo AS {<null>;String;Interzo}

```

Diagramma UML delle classi ristrutturato



Breve descrizione delle scelte effettuate durante la ristrutturazione

Hc inserito (non riportati) in ogni classe è rimasta 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)

Sono emersi solo 2 nuovi vincoli

[V. CLIENTE.F. N°IMP] T

✓ CF, U, A

(CLIENTE.FCO (CF) n CLIENTE IMPRESA (U) n APPROVVIMENTO (A) n AB_CLIENTE (A, CF) \rightarrow AB_CLIENTE (A, CF))

[V. CLIENTE.I. N°FS]

✓ CF, U, A

(CLIENTE.FCO (CF) n CLIENTE IMPRESA (U) n APPROVVIMENTO (A) n AB_CLIENTE (A, CF) \rightarrow AB_CLIENTE (A, CF))

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 UML delle classi ristrutturato.

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

1 Relazione	<u>NAZIONE</u> (nome)	Derivante da: classe associazione (cerchiare)
--------------------	-----------------------------	--

Attributi	<u>Nome</u>							
-----------	-------------	--	--	--	--	--	--	--

Domini	<u>STRANGA</u>							
--------	----------------	--	--	--	--	--	--	--

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

2 Relazione	<u>CITTÀ</u> (nome)	Derivante da: classe associazione (cerchiare)
--------------------	---------------------------	--

Attributi	<u>ID</u>	<u>Nome</u>	<u>NAZIONE</u>					
-----------	-----------	-------------	----------------	--	--	--	--	--

Domini	<u>SERIAL</u>	<u>STRANGA</u>	<u>STRANGA</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: NAZIONE REFERENCES NAZIONE(na)

La relazione accorda le relazioni che implementano le seguenti associazioni: ...UT-NAZ.....

3 Relazione	<u>UTENTE</u> (nome)	Derivante da: classe associazione (cerchiare)
--------------------	----------------------------	--

Attributi	<u>ID</u>	<u>Nome</u>	<u>Cognome</u>	<u>DATA_Nascita</u>	<u>Indirizzo</u>	<u>Email</u>	<u>CITTA</u>	<u>POSTIZIONE</u>
-----------	-----------	-------------	----------------	---------------------	------------------	--------------	--------------	-------------------

Domini	<u>SERIAL</u>	<u>STRANGA</u>	<u>STRANGA</u>	<u>DATO</u>	<u>Indirizzo</u>	<u>Email</u>	<u>Integ</u>	<u>Integ</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 DEF CITTA(CIT)

FK: APPARTAMENTO DEF APPARTAMENTO(CIT)

FK: POSIZIONE DEF POSIZIONE(ID)

La relazione accorda le relazioni che implementano le seguenti associazioni: UT-CIT, UT-POS, AB-UTENTE....

4 Relazione	<u>ACCESSO</u> (nome)	Derivante da: classe associazione (cerchiare)
--------------------	-----------------------------	--

Attributi	<u>ID</u>	<u>DATA</u>	<u>ENTRATA</u>	<u>USCITA</u>	<u>UTENTE</u>	<u>POSTIZIONE</u>		
-----------	-----------	-------------	----------------	---------------	---------------	-------------------	--	--

Domini	<u>SERIAL</u>	<u>DATO</u>	<u>TIME</u>	<u>TIME</u>	<u>Integ</u>	<u>Integ</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):

CHECK (ENTRATA < USCITA IF USCITA IS NOT NULL)

FK: POSTIZIONE DEF POSTIZIONE(ID)

FK: UTENTE DEF UTENTE(ID)

La relazione accorda le relazioni che implementano le seguenti associazioni: ...UT-ACC, ...ACC-POSTI.....

5 Relazione	<u>POSTIZIONE</u> (nome)	Derivante da: classe associazione (cerchiare)
--------------------	--------------------------------	--

Attributi	<u>ID</u>	<u>UTENTE</u>						
-----------	-----------	---------------	--	--	--	--	--	--

Domini	<u>SERIAL</u>	<u>Integ</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 DEF UTENTE(ID)

La relazione accorda le relazioni che implementano le seguenti associazioni: ...UT-POSTI.....

6 Relazione <u>ABBRACCIAZIONE</u> (nome)					Derivante da: <u>classe</u> associazione (cerchiare)			
Attributi	<u>ID</u>	INIZIO	TIPO	<u>CLIENTE</u> * / <u>CLIENTE</u> /*				
Domini	<u>SERIAL</u>	<u>DATETIME</u>	<u>STRINGA</u>	<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: TIPO DEF TIPO(CLIENTE) (U. INCLUSIONE) : ABBRACCIAZIONE (n) inciso in UTENSILE (ABBRACCIAZIONE)

FK: CLIENTE F DEF CLIENTE(FKSO (n)) FK: CLIENTE DEF CLIENTE(IMPRESA(n)) - ABBRACCIAZIONE (n) inciso in AB3_GD (ABBRACCIAZIONE)

La relazione accorda le relazioni che implementano le seguenti associazioni: AB3_TIP_Q., AB_CLIENTE_F,

AB_CLIENTE_I

7 Relazione <u>TIPOLOGIA</u> (nome)					Derivante da: <u>classe</u> associazione (cerchiare)			
Attributi	<u>NOME</u>	PIVA	DURATA_GIORNI	STIPULANTE_NA	STIPULANTE_A	MAX_UTENTI		
Domini	<u>STRINGA</u>	<u>REAL</u>	<u>INTERO</u>	<u>DATETIME</u>	<u>DATETIME</u>	<u>INTERO</u>		

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

8 Relazione <u>CLIENTE_FESCO</u> (nome)					Derivante da: <u>classe</u> associazione (cerchiare)			
Attributi	<u>ID</u>	NOME	Cognome	DATA_N	PIVA*			
Domini	<u>SERIAL</u>	<u>STRINGA</u>	<u>STRINGA</u>	<u>DATE</u>	<u>PIVA</u>			

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

9 Relazione <u>CLIENTE_IMPRESA</u> (nome)					Derivante da: <u>classe</u> associazione (cerchiare)			
Attributi	<u>PIVA</u>	RAGONE_SOCIE						
Domini	<u>PIVA</u>	<u>RAGONE_SOCIE</u>						

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

10 Relazione <u>SCENZA</u> (nome)					Derivante da: <u>classe</u> associazione (cerchiare)			
Attributi	<u>NAME</u>	DESC	<u>PIVA</u>					
Domini	<u>STRINGA</u>	<u>STRINGA</u>	<u>REAL</u>					

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

11 Relazione	<u>AB-SEGU.....</u> (nome)	Derivante da: classe associazione (cerchiare)
Attributi	<u>APPONIMENTO</u> <u>SERVIZIO</u> <u>SCARICO</u> <u>SOCIALE</u> <u>GRANDE</u>	
Domini	<u>Intero</u> <u>STRANIERO</u> <u>SERVIZIO</u> <u>Intero</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: APPONIMENTO DEF APPONIMENTO (id)

FK: SERVIZIO DEF SERVIZIO (name)

La relazione accorda le relazioni che implementano le seguenti associazioni:

12 Relazione	<u>UT-UTENZE..</u> (nome)	Derivante da: classe associazione (cerchiare)
Attributi	<u>ID</u> UTENZE <u>SERVIZIO</u> QUANTITA' <u>WIZZ</u> <u>FINE</u> <u>DATA</u>	
Domini	<u>SERIAL</u> <u>INTEGER</u> <u>STRANIERA</u> <u>Intero</u> <u>TIME</u> <u>TIME</u> <u>DATE</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: UTENZE DEF UTENZE (id)

FK: SERVIZIO DEF SERVIZIO (name)

La relazione accorda le relazioni che implementano le seguenti associazioni: UT-UTENZE.. , UT-SERVIZIO ..

13 Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi		
Domini		

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

14 Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi		
Domini		

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

15 Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi		
Domini		

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

16 Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi	
Domini	

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

17 Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi	
Domini	

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

18 Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi	
Domini	

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

19 Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi	
Domini	

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

20 Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi	
Domini	

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

Ulteriori vincoli esterni

Per ogni ulteriore vincolo esterno (non ancora espresso perché non definibile mediante vincoli di chiave, foreign key, ennupla, dominio, inclusione), progettare un trigger che lo implementi, definendo: (a) gli eventi da intercettare (inserimento, modifica, eliminazione di ennupla); (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.

T. CLIENTE F. NO IMP

O INSERIMENTO o MODIFICA IN ABONAMENTO

o PRE-OPERAZIONE

```
is_Error := (exists (SELECT LF.ID
                      FROM ABONAMENTO A, CLIENTE_FISICO CF, CLIENTE_IMMOSA CI
                     WHERE A.CLIENTEF = CF.ID AND A.CLIENTEI = CI.ID AND
                           CF.ID < CI.ID AND NEW.ID = A.ID))
```

```
if is_Error:
    GENERA_ERRORE
    REVERT
```

```
else:
    COMM
```

T. TIPOLO OA. MAX CLIENTI

O INSERIMENTO o MODIFICA IN ABONAMENTO

o PRE-OPERAZIONE

```
is_Error := (exists (SELECT (T.MAXCLIENTI < COUNT(A.ID))
                      FROM ABONAMENTO A, TIPOLO GA T
                     WHERE A.ID = NEW.ID AND A.TIPO = T.NOME))
```

```
if is_Error:
    GENERA_ERRORE
    REVERT
```

```
else:
    COMM
```

T. ABONAMENTO. INVAZ

O INSERIMENTO o MODIFICA IN ABONAMENTO

o PRE-OPERAZIONE

```
is_VAULT := (not exists (SELECT A.ID
                           FROM ABONAMENTO A, TIPOLO GA T
                          WHERE A.ID = NEW.ID AND A.TIPO = T.NOME
                            AND (A.WIZARD < T.SUPERVISORE OR A.WIZARD > T.SUPERVISORE)))
```

```
if is_VAULT:
    COMM
```

```
else:
    GENERA_ERRORE
    REVERT
```

Risposta alla Domanda 7 (segue)

T. Accesso. RESTAURAZIONE

- INSERIMENTO o MODIFICA IN ACCESSO

- PER OPERAZIONE

$\text{is_Endr} := \exists \text{S} \left(\begin{array}{l} \text{SELECT A.ID} \\ \text{FROM Accesso A, Utente U, Postazione P} \\ \text{WHERE NEW.ID = A.ID AND A.Utenze > U.ID AND} \\ \text{A.Postazione = P.ID AND P.Utenze <= U.ID} \end{array} \right)$

if isEndr:
Genera errore
return
else:
comm

T. ACCESSO. CONTINUITÀ

- INSERIMENTO o MODIFICA IN ACCESSO

- PER OPERAZIONE

$\text{is_Endr} := \exists \text{S} \left(\begin{array}{l} \text{SELECT A.ID} \\ \text{FROM Accesso} \\ \text{WHERE A.Gruppo = A.Utenza} \end{array} \right)$

if isEndr:
comm

else:
Genera errore
return

else:
comm

$\forall A_1, U_1, A_2 \in$
 $\{ \text{ABONAMENTO}(A_1, INIZIO(A_1) \wedge UTENZA(U_1)) \}$
 $\wedge \text{AB}_1 \text{UT}(A_1, U_1) \wedge \text{ACCESSO}(A_1) \wedge \text{UTACC}(U_1, A_2)$
 $\wedge \text{ENTRATA}(AC, E) \} \rightarrow \text{ICE}$

T. ACCESSO. ENTRABONAMENTO

- INSERIMENTO o MODIFICA IN ACCESSO

- PER OPERAZIONE

$\text{is_Endr} := \exists \text{S} \left(\begin{array}{l} \text{SELECT AC.ID} \\ \text{FROM Accesso AC, Utente U, Abbonamento AB} \\ \text{WHERE NEW.ID = AC.ID AND AC.Utenze = U.ID AND} \\ \text{U.Abbonamento = AB.ID AND AC.Data < AB.Inizio} \end{array} \right)$

if isEndr:
Genera errore
return

else:
comm

Domanda 8 (30 minuti; 45 minuti al massimo) Proseguire la fase di progettazione dell'applicazione producendo le specifiche realizzative delle operazioni di classe e/o use-case definite per modellare i requisiti contrassegnati dalla barra laterale della specifica dei requisiti.

In particolare, per ogni operazione definire la segnatura, in termini di nome dell'operazione, nomi e dominio SQL degli argomenti, dominio SQL dell'eventuale valore di ritorno, e un algoritmo in pseudo-codice con SQL immerso che verifichi le precondizioni e garantisca il raggiungimento delle postcondizioni definite in fase di Analisi. Specificare, per ogni operazione, se debba essere implementata nel DBMS o nel back-end.

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

Risposta

OPERAZIONI DI CLASSE

CLASSE ABONAMENTO

-CALCOLA_SALDO (in:DataAbbo, fin:DataAbbo, AB:integer) : REALE

CREAIE Funzione

Q = $\text{Select} \left(\left(S.\text{PREZZO UNITARIO} \cdot C1 - AS.\text{SCARICO} \right) \cdot \left(\text{Sum}(US.\text{QUANTITA}) - AS.\text{SOLTI GRATIS} \right) \right) \text{ AS TOT}$

From ABONAMENTO A, AB_SERV AS, SERVIZIO S, UTENTE U, USOSERVIZIO US

WHERE A.ID = AB AND AS.ABONAMENTO = A AND AS.SERVIZIO = S.NOME

AND U.ABONAMENTO = A.ID AND US.UTENTE = U.ID AND US.SERVIZIO = S.NOME
AND IN <= US.GIORNO AND OS.GIORNO <= FIN

RESULT = Q

CLASSE ACCESSO

-TOLACCESSI (D:DATA, A:integer) : Intero

CREAIE Funzione

Q = $\text{Select COUNT(DISTINCT A.ID)} \text{ From ACCESSO A WHERE A.ID = A AND A.DATA = D}$

RESULT = Q

Risposta alla Domanda 8 (segue) USE CASE

SPESA (I: PIVA, IN: DATAINIZIA, FIN: DATAFINE): REALE

MEMORIZZA IN Q LA QUERY

SELEZIONA SUM(CALCOLA_SPESA(I, IN, A.ID))

FROM APPARTAMENTO A, CLIENTE INDICES C1

WHERE A.CODICEFISCALE NOT NULL AND A.CODICEFISCALE = C1.PIVA AND C1.ID = 1

RESULT = Q

STATISTICHE ([I: IN, TIME, FIN: TIME] [L: n]) : ([IN: TIME, FIN: TIME], N) [L: n]

TABELLA(GLOBO, (D: DATE))

WITH RECURSIVE GLOBO(D) AS (

SELECT DATE_TRUNC('month', NOW()) AS Mese

UNION ALL

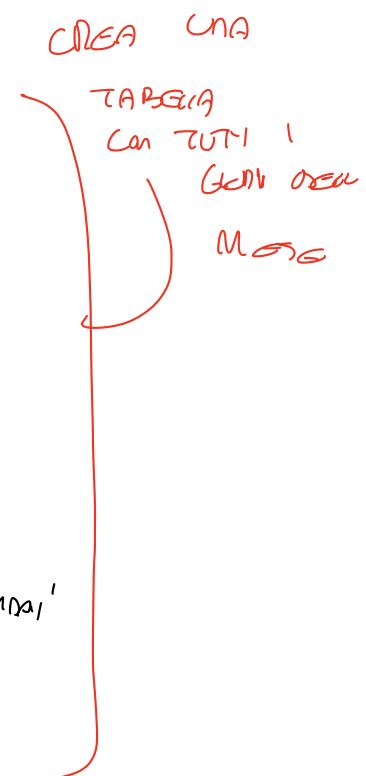
SELECT Mese + INTERVAL '1 DAY'

FROM GLOBO

WHILE

Globo < DATE_TRUNC('month', now()) + interval '1 Month' - interval '1 Day'

INSERT INTO TABELLA (GLOBO)



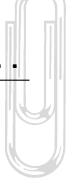
MEMORIZZA IN Q LA QUERY

SELECT EXTRACT('DAY', TG.D) AS GLOBO, (I.IN, I.FIN), (COUNT(A.ID) / GLOBO) AS NUM

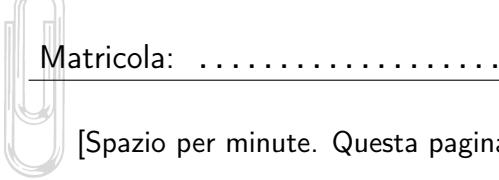
FROM ACCESSO A

WHERE A.ENTRATA > I.INIZIO AND A.USCITA < I.FINE

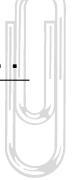
Tempo totale stimato per svolgere questa prova: 180 minuti (tempo totale concesso: 300 minuti).
[Spazio per minute. Questa pagina non sarà valutata a meno che non sia puntata da pagine precedenti.]



[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.]