



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), *includere* 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



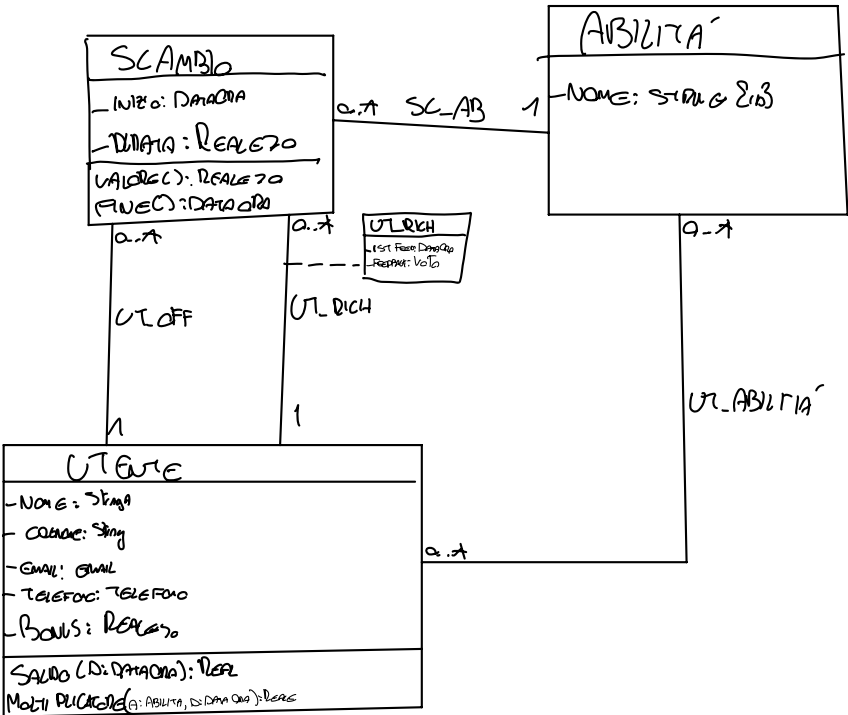
Risposta alla Domanda 1 (segue)

**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.





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  $assoc/2$  per ogni associazione binaria  $assoc$ .  
Semantica di  $assoc(c_1, c_2)$ :  $(c_1, c_2)$  è una istanza di  $assoc$ .
  - Un simbolo di predicato  $attr/2$  per ogni attributo  $attr$  di entità  
Semantica di  $attr(c, v)$ : uno dei valori dell'attributo  $attr$  dell'istanza  $c$  è  $v$ .
  - Un simbolo di predicato  $attr/3$  per ogni attributo  $attr$  di associazione binaria.  
Semantica di  $attr(c_1, c_2, v)$ : uno dei valori dell'attr.  $attr$  del link  $(c_1, c_2)$  è  $v$ .
  - Un simbolo di predicato  $op/(n + 2)$  per ogni operazione di classe ad  $n$  argomenti.  
Semantica di  $op(c, arg_1, \dots, arg_n, v)$ : uno dei valori di ritorno di  $op$ , quando invocata sull'istanza  $c$  e con argomenti  $arg_1, \dots, arg_n$  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 adesso/0, interpretato come il valore del dominio DataOra che rappresenta l'istante corrente.

Risposta

<div>1 Tipo: <b>Classe</b>   Associazione (cerchiare)</div> <div>Nome: <u>UTENTE</u></div> <div>Operazioni, vincoli: <u>[U.SALDO.NOBESSE]</u> T <math>\forall u, s, d</math> <math>[UTENTE(u) \wedge DATA(s) \wedge SALDO(u, d, s)] \rightarrow s \geq 0</math> <u>[U.UTENTE.OFFNOBILIA]</u> T <math>\forall u, s</math> <math>[UTENTE(u) \wedge SALDO(s) \wedge OFF(u, s)] \rightarrow \neg UT\_NOB(u, s)</math></div>	<div>2 Tipo: <b>Classe</b>   Associazione (cerchiare)</div> <div>Nome: <u>UTENTE</u></div> <div>Operazioni, vincoli: <u>[V.UTENTE.AAABILIA]</u> T <math>\forall u, s, A</math> <math>[UTENTE(u) \wedge SALDO(s) \wedge ABILIA(u) \wedge UT\_OFF(u, s) \wedge SALDO(s, A)] \rightarrow UT\_ABILIA'(u, A)</math></div>
---	--

<div><div>3</div><div>Tipo: <b>Classe</b>   <b>Associazione</b> (cerchiare)</div><div>Nome: SCAMBIO</div><div>Operazioni, vincoli:</div><div><div>[V.SCAMBIO.FEEDBACK]</div><div><math>\forall S_1, D, F, U</math></div><div><math>[SCAMBIO(S) \wedge INIZIO(S, D) \wedge FINE(U, D, F) \wedge UTENTE(U) \wedge FEED(S, U, F)] \rightarrow (F \neq F)</math></div></div></div>	<div><div>6</div><div>Tipo: <b>Classe</b>   <b>Associazione</b> (cerchiare)</div><div>Nome: UTENTE</div><div>Operazioni, vincoli:</div><div><div>[V.UTENTE.NOCONTENDRANGE]</div><div><math>\forall U_1, S_1, P_1, U_2, D_1, F_1, U_3, D_2, F_2</math></div><div><math>[UTENTE(U_1) \wedge SCAMBIO(S_1) \wedge INIZIO(S_1, D_1) \wedge FINE(U_1, D_1, F_1) \wedge UTENTE(U_2) \wedge SCAMBIO(S_2) \wedge INIZIO(S_2, D_2) \wedge FINE(U_2, D_2, F_2) \wedge S_1 \neq S_2 \wedge U_1 \neq U_2 \wedge U_1 \neq U_3 \wedge U_2 \neq U_3] \rightarrow</math></div><div><math>\neg (U_1 &gt; P_2 \vee U_2 &gt; P_1)</math></div></div></div>
<div><div>4</div><div>Tipo: <b>Classe</b>   <b>Associazione</b> (cerchiare)</div><div>Nome: .....</div><div>Operazioni, vincoli:</div></div>	<div><div>7</div><div>Tipo: <b>Classe</b>   <b>Associazione</b> (cerchiare)</div><div>Nome: .....</div><div>Operazioni, vincoli:</div></div>
<div><div>5</div><div>Tipo: <b>Classe</b>   <b>Associazione</b> (cerchiare)</div><div>Nome: .....</div><div>Operazioni, vincoli:</div></div>	<div><div>8</div><div>Tipo: <b>Classe</b>   <b>Associazione</b> (cerchiare)</div><div>Nome: .....</div><div>Operazioni, vincoli:</div></div>

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

## OPERAZIONI DI CLASSE

### -CLASSE SCAMBIO

**OFNAC(): DATA ORA**

OPRE: NESSUNA

° POST:  $M_{11} = N_{cur}$

SIA  $I, D \text{ INIZIO}(th_{13}, i) \wedge \text{DATA}(th_{13}, D)$

Result =  $1 + D$

**Q VALORE(): REALE 70**

PRE: NESSUNA

POST:  $M_{11} = N_{cur}$

SIA  $D \text{ DATA}(th_{13}, D)$

SIA  $G \text{ INIZIO}(th_{13}, G)$

SIA  $A \text{ SCALAB}(th_{13}, A)$

SIA  $U \text{ UT OFF}(G, th_{13})$

SIA  $M \text{ MOLTIPLICAZIONE}(U, A, G)$

Result =  $D \cdot M$

### CLASSE UENTE

**CMOLTIPLICARE(A: ARBITRA, D: DATA ORA): REALE 70**

PRE: NESSUNA

POST:  $M_{11} = N_{cur}$

SIA  $S = \left\{ (S, PB) \mid \begin{array}{l} \text{SCAMBIO}(S) \wedge \text{UT OFF}(th_{13}, S) \wedge \text{ADESSO}(A) \\ \wedge \text{FINE}(S, F) \wedge \text{IF} \wedge \{ \text{UT BUTTARE}(U) \wedge \text{UT DEC}(U, 2) \wedge \text{FEEDBACK}(U, S, PB) \} \\ \wedge \text{ARBITRA}(A) \wedge \text{SCALAB}(S, A) \end{array} \right\}$

$$\text{MEDIA} = \frac{\sum PB}{|S|}$$

Result =  $0$  se  $|S| = 0$

-  $1 + \text{median}(\text{last} + \text{MEDIA})$

**SALNO(D: DATA ORA): REALE 70**

OPRE: NESSUNA

° POST:  $M_{11} = N_{cur}$

SIA  $X = \text{BONUS}(th_{13}, X)$

SIA  $R = \left\{ (S, V) \mid \begin{array}{l} \text{UT DEC}(th_{13}, S) \wedge \exists G \text{ INIZIO}(S, G) \wedge G \leq D \wedge \\ \text{VALORE}(S, V) \wedge \exists F \text{ FINE}(S, F) \wedge \text{ADESSO}(A) \wedge \text{FLA} \end{array} \right\}$

$$\text{TOT SPET} = \sum_{G \leq D} V$$

SIA  $O = \left\{ (S, V) \mid \begin{array}{l} \text{UT OFF}(th_{13}, S) \wedge \exists G \text{ INIZIO}(S, G) \wedge G \leq D \wedge \\ \wedge \text{VALORE}(U, S) \wedge \exists F \text{ FINE}(S, F) \wedge \text{ADESSO}(A) \wedge \text{FLA} \end{array} \right\}$

Risposta alla Domanda 2 (segue)

$$\text{TOT DC} = \sum_{\text{SPES}} v$$

$$\text{RESULT} = (\text{TOT DC} + x) - \text{TOT SPES}$$

TIP DI DATO

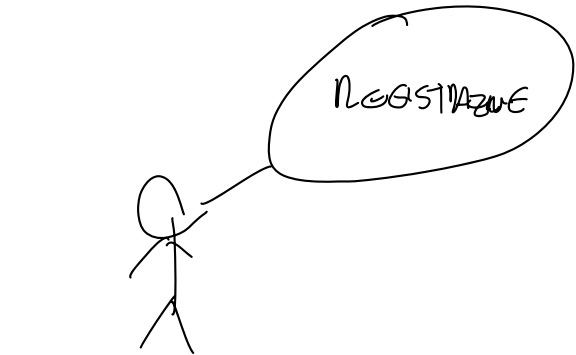
Voto: O..S

EMAIL: STRING SECONDO REGEX

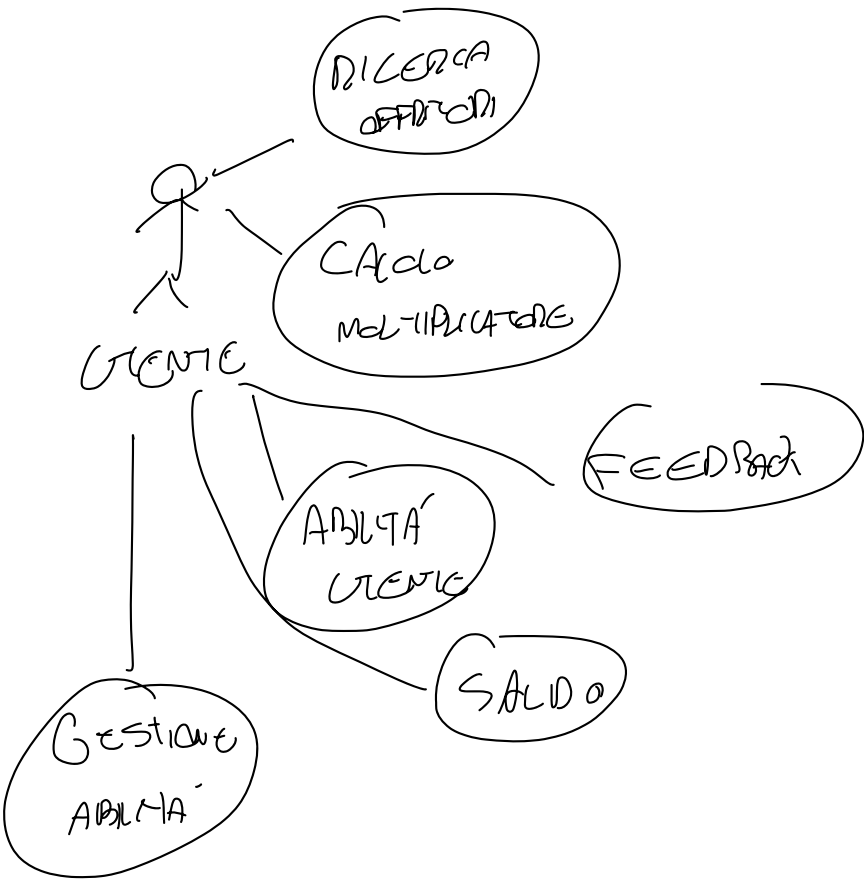
TELEFONO: STRING SECONDO REGEX

**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



utente Non REG





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 REGISTRAZIONE

REGISTRAZIONE(U: STRING, C: STRING, E: EMAIL, T: TELEFONO, B: REGIONE): UTENTE

#### USE CASE RICERCA OFFERTORI

RICERCA(A: ABILITA', m: REGIONE): {UTENTE} [C..T]

#### USE CASE CALCOLO MULTIMILIATORE

MULTIPLICAZIONE(CO: DATAORA, A: ABILITA'): REGIONE

#### USE CASE ABILITA' UTENTE

ABILITA' UTENTE(U: UTENTE): {ABILITA'} [C..T]

#### USE CASE GESTIONE ABILITA'

- NUOVA ABILITA' (U: STABO): ABILITA'
- ELIMINA ABILITA' ( ): {ABILITA'} [C..T]
- RIMUOVA ABILITA' (A: ABILITA'):

#### USE CASE FEEDBACK

FEEDBACK(S: SAMPLO, U: UTENTE): UTENTE

#### USE CASE SALDO

SALDO(): REAL



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

**RICERCA (A: ABILITA', m: REAG): [UTENTE] [C.A.]**

PRE CONDIZIONI: NESSUNA

OPERAZIONI AUSILIARIE

MODA-FEEDBACK (U: UTENTE): REAG

PRE: NESSUNA

POST:  $M_{in} = M_{out}$

$R = \left\{ (C, S, F) \mid \begin{array}{l} \text{SLAMPO}(S) \wedge \text{UTOFF}(U, S) \wedge (\exists' U' \text{ UTENTE}(U') \wedge \text{CRASH}(U', S)) \\ \text{ABILITA}(A) \wedge (\exists F \text{ AVG}(C, A) \wedge \text{REAG}) \wedge \text{FEEDBACK}(U, S, F) \end{array} \right\}$

$$\text{RESULT} = \sum_{(C, S, F)} F = \frac{1}{|R|}$$

ABILI CAMBIA:  $M_{in} = M_{out}$

SIA  $R = \left\{ U \mid \text{UTENTE}(U) \wedge \text{UTABILITA}(U, A) \wedge \exists k \text{ MODA-FEEDBACK}(U, k) \wedge k \neq m \right\}$

RESULT = R

**MOLTEPLICAZIONE (U: DATAGRA, A: ABILITA'): REAG**

SI A U L'ATTORE CHE INCA U USE CASE

OPDE:  $\text{UTATI}(U, A)$

OPDE:  $M_{in} = M_{out}$

RESULT = MOLTEPLICAZIONE (U, A, D)



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 multivalore 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.

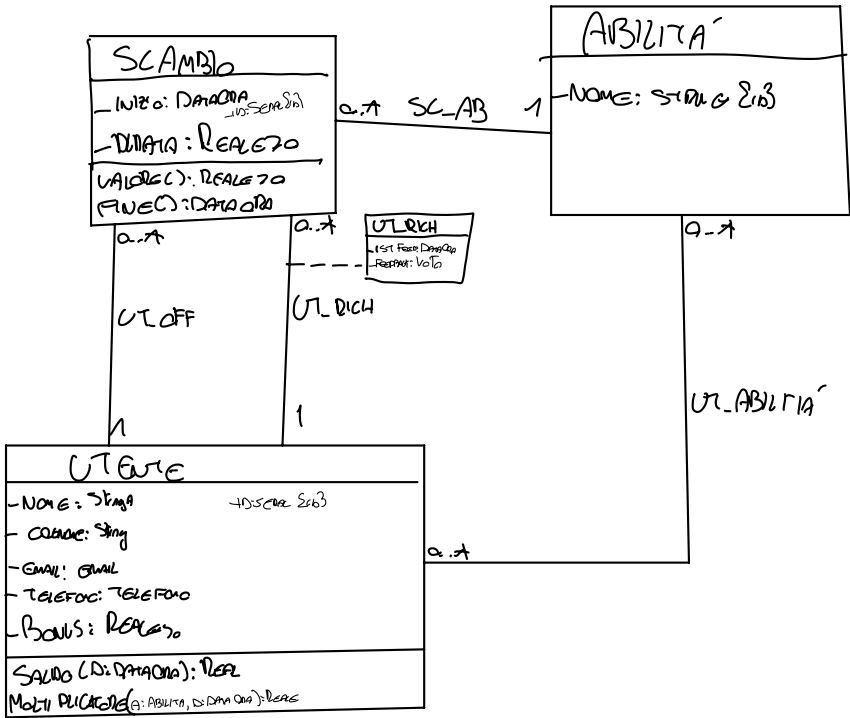
Descrivere brevemente le principali scelte effettuate.

DBMS da utilizzare .... Post SQL .....

Corrispondenza tra tipi di dato concettuali e domini supportati dal DBMS

CREATE DOMAIN String AS VARCHAR (check VALUE IS NOT NULL)  
 CREATE DOMAIN Intero AS Integer (check VALUE > 0)  
 CREATE DOMAIN Intero70 AS Integer (check VALUE < 70)  
 CREATE DOMAIN Reale70 AS REAL (check VALUE < 70)  
 CREATE DOMAIN Reale70 AS REAL (check VALUE < 70)  
 CREATE DOMAIN Voto AS Integer (check VALUE > 0 AND VALUE < 5)  
 CREATE DOMAIN EMAIL AS VARCHAR (check VALUE ~ REGEX)  
 CREATE DOMAIN TEL AS VARCHAR (check VALUE ~ REGEX)

Diagramma UML delle classi ristrutturato



Breve descrizione delle scelte effettuate durante la ristrutturazione

UGACE

**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)

UGACE

[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 UML delle classi ristrutturato.

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

1

Relazione UTENTE..... (nome)

Derivante da: classe | associazione (cerchiare)

Attributi	<u>ID</u>	<u>NAME</u>	<u>COGNOME</u>	<u>EMAIL</u>	<u>TELEFONO</u>			
Domini	SERIAL	STRINGA	STRINGA	EMAIL	TEL			

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 associazioni: .....

2

Relazione ABUSO..... (nome)

Derivante da: classe | associazione (cerchiare)

Attributi	<u>NAME</u>							
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 associazioni: .....

3

Relazione UTENTE..... (nome)

Derivante da: classe | associazione (cerchiare)

Attributi	<u>UTENTE</u>	<u>ABUSO</u>						
Domini	Integer	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: UTENTE DEF UTENTE(n)  
FK: ABUSO DEF ABUSO(uno)

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

4

Relazione SCAMPO..... (nome)

Derivante da: classe | associazione (cerchiare)

Attributi	<u>ID</u>	<u>DATA</u>	<u>NUMERO</u>	<u>DESCRIZIONE</u>	<u>ABUSO</u>			
Domini	SERIAL	DATE	NUMERO	DESCRIZIONE	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: UTENTE DEF UTENTE(n)  
FK: ABUSO DEF ABUSO(uno)

La relazione accorpa le relazioni che implementano le seguenti associazioni: UTENTE DEF SCAMPO.....

5

Relazione UTENTE..... (nome)

Derivante da: classe | associazione (cerchiare)

Attributi	<u>UTENTE</u>	<u>SCAMPO</u>	<u>FEEDBACK</u>	<u>ISFEEDBACK</u>				
Domini	Integer	Integer	VOLO	BOOLEANO				

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(n)  
FK: SCAMPO DEF SCAMPO(n)

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

6 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 accorpa le relazioni che implementano le seguenti associazioni: .....

7 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 accorpa le relazioni che implementano le seguenti associazioni: .....

8 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 accorpa le relazioni che implementano le seguenti associazioni: .....

9 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 accorpa le relazioni che implementano le seguenti associazioni: .....

10 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 accorpa le relazioni che implementano le seguenti associazioni: .....



11

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 accorpa le relazioni che implementano le seguenti associazioni: .....

12

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 accorpa le relazioni che implementano le seguenti associazioni: .....

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 accorpa 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 accorpa 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 accorpa le relazioni che implementano le seguenti associazioni: .....

Versione 2024-06-05.UML]

### Ulteriori vincoli esterni

Per ogni ulteriore vincolo esterno (non ancora espresso perché non definibile mediante vincoli di chiave, foreign key, enunpla, dominio, inclusione), progettare un trigger che lo implementi, definendo: (a) gli eventi da intercettare (inserimento, modifica, eliminazione di enunple); (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. SALINC. NO RAZZO

○ INSERIMENTO ○ MODIFICA IN UTENTE

○ PRIMA OPERAZIONE

```
IS ERRORE := (EXISTS (SELECT U.ID
                     FROM UTENTE U
                     WHERE U.ID = NEW.ID AND SALINC(U, NAME)) > 0)
```

IF ISERRORE: SEVERA ERRORE  
RAZZO

ESGO: COMMI

#### T. SCAMBIO DEFNODICH

○ INSERIMENTO ○ MODIFICA IN SCAMBIO

○ PRIMA OPERAZIONE

```
IS ERRORE := (EXISTS (SELECT S.ID
                     FROM UTENTE U, SCAMBIO S, UTENTE V
                     WHEN S.ID = NEW.ID AND SCAMBIO = U.ID
                     AND U.SCAMBIO = S.ID AND U.UTENTE = SCAMBIO))
```

IF ISERRORE: SEVERA ERRORE  
RAZZO

ESGO: COMMI

## Risposta alla Domanda 7 (segue)

T. UTENTE.ABILITA'

O INFERMIER O MONITOR IN UTENTE

O PREORDINAZIONE

```

is valid: (EX1 (SELECT U.ID
FROM UTENTE U, STAMPA S, ABILITA' A, CLASSTIA' UA
WHERE NEW.ID = U.ID AND UA.UTENTE = U.ID AND S.REFERE = U.ID
AND S.ABILITA' = UA.ABILITA'))

```

```

IF is valid;
  COMMIT

```

```

ELSE:
  GENERA ERRORI
  ROLLBACK

```

**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 SCAMBIO

OFINDE(S:integer): DATACONA

CREATE FUNCTION

Q = ( SELECT (S.INTRAT + S.DATATA) )  
FROM SCAMBIO S  
WHERE S.ID = S

RESULT = Q

Q VALORE(S:integer): REAL70

CREATE FUNCTION

Q = ( SELECT (S.DATATA \* U.MULTIPLOCAND(U.ID, S.INTRAT, S.DATATA)) )  
FROM SCAMBIO S, UTENTE U  
WHERE S.ID = S AND S.OPERATE = U.ID

RESULT

OMMULTIPLOCAND(U:integer, G: DATETIME, A: STIMICA): REAL70

CREATE FUNCTION

Q = ( SELECT (G + ARCTAN(COS(MEDIA(U, A, G)) AS MOL) ) )  
FROM UTENTE U WHERE U.ID = U

RESULT = Q

## Risposta alla Domanda 8 (segue)

**MEDIA (U: Integer, A: STRINGA, D: DATADIA)**

CREATE FUNCTION

G: (EXISTS (SELECT \*  
FROM UENTE U, ULAB UA  
WHERE U.ID=U AND UA.UTENTE=U.ID AND UA.AUTORE=A))

IF E: FASE:

GENERARE

ELSE:

Q = (SELECT AVG (UR.FEEDBACK) AS MEDIA  
FROM UENTE U, SLAMB S, ULAB UR  
WHERE U.ID=U AND S.UTENTE=U.ID AND S.AUTORE=A  
AND S.FINE < NOW() AND UR.SLAMB=S.ID)

RESULT = Q

**SALVO (D: DATETIME, U: Integer): DECREMENTO**

CREATE FUNCTION

Q:=  
WITH SPES1 AS (SELECT SUM (S.VALE(S.ID)) AS SPES1  
FROM SLAMB S, UENTE U, ULAB UR  
WHERE UR.SLAMB=S.ID AND U.ID=U AND UR.UTENTE=U.ID)

WITH GUADAGN1 AS (SELECT SUM (S.VALE(S.ID)) AS GUAD1  
FROM SLAMB S, UENTE U  
WHERE S.UTENTE=U.ID AND U.ID=U)

SELECT ((GUADAGN1.GUAD + U.PONTS) - (SPES1.SPES1)) AS SALVO

FROM UENTE U, GUADAGN1, SPES1

WHERE U.ID=U

RESULT = Q - SALVO

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.]

## USE CASE

**RICERCA(A: stringa, m: Page): [utente] [c]**

SALVA IN Q LA QUERY SQL

SELEZIONA C.M

FROM UTENTE U, UT.ATTIVA UA

WHERE UA.ATTIVA = A AND UA.UTENTE = U.ID  
AND U.MEMB(C.M, A, NOW()) > 0

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.]



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