

**Esame BD2.Esame.Risposte.ER – Modulo risposte prova scritta**

**Dati dello studente e dell'esame**

Cognome e nome: Caso Marco Matricola: .....

Data: 12/06/2014 .....

Corso di laurea e canale di appartenenza:

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

Firma di un membro della Commissione per  
avvenuta identificazione:

.....

**Rinuncia alla prova**

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

Time Bank



Questo modulo è ottimizzato per la stampa fronte-retro

# 1 Analisi concettuale

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

Risposta

moltiplicatore (di: calcolato in un dato istante

01. Utente — 1 — 2. Abilita  
 01.2 nome 2.1 utente che la fornisce  
 01.3 Saldo() → Scambi offerti - richiesti  
 01.4 cognome  
 01.1 anzil  
 01.5 cell  
 2. abilità erogate  
 scambi  
 saldo bonus: x
03. Scambio  
 3.1 utente richiedente  
 3.2 abilità  
 3.3 utente fornitore  
 03.4 DataOra  
 03.5 Durata } il valore dipende  
 3.6 valore } dalla durata  
 03.7 Feedback [0..1]

Saldo(d:lst) = saldo dell'utente nell'ist. d

Utente — 1 — Abilita  
 Offre  
 molte

V. utente richiede scambio  
 se non è in rosso

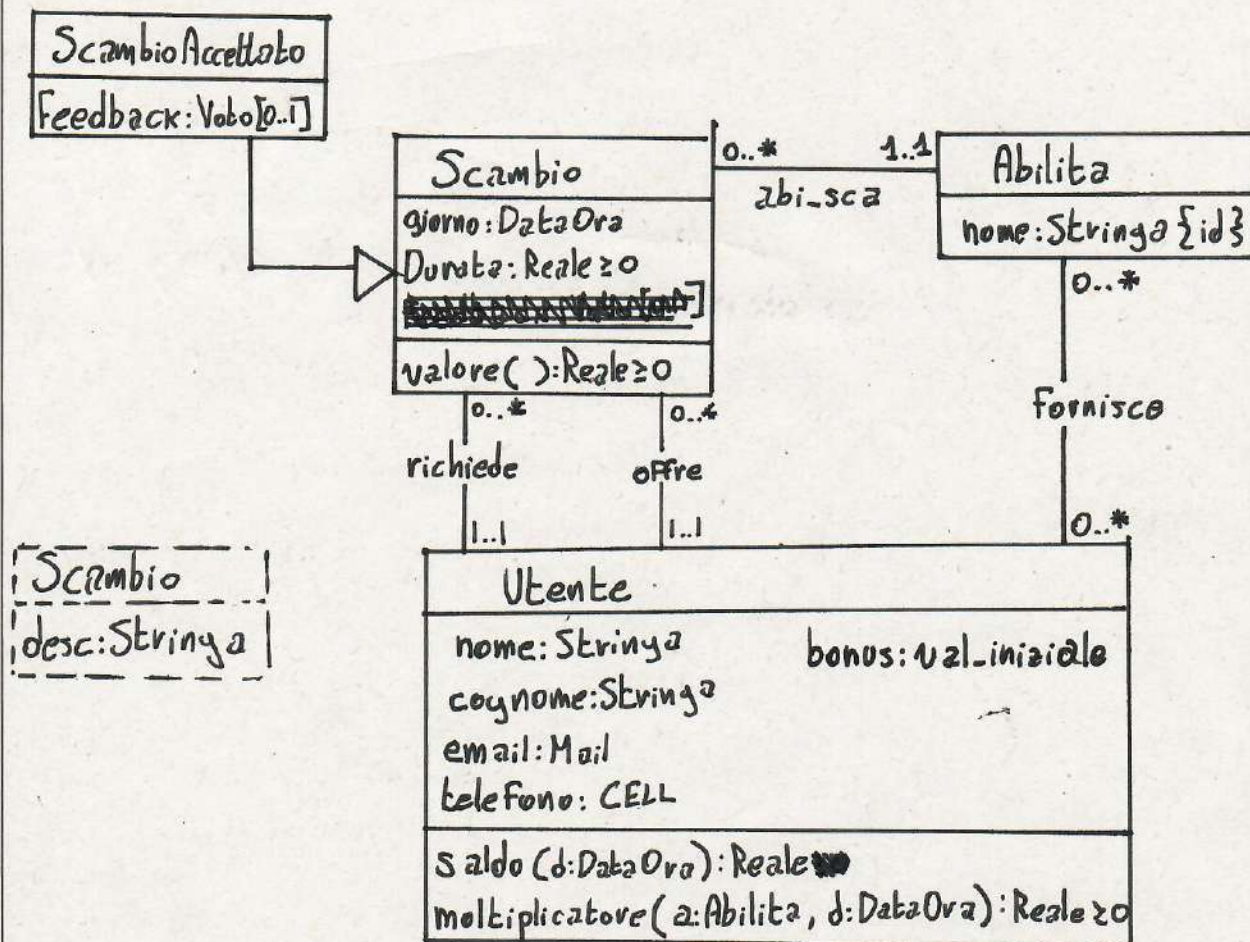


**Domanda 2 (45 minuti; 75 minuti al massimo)** Proseguire la fase di Analisi Concettuale dei requisiti, producendo un diagramma ER concettuale per l'applicazione, il dizionario dei dati ed eventuali vincoli esterni.

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.



Un Utente può:

- richiedere ed offrire contemporaneamente
- richiedere 2 scambi contemporaneamente

Non può:

- offrire 2 scambi contemporaneamente



**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: <u>Entità</u>   Relationship (cerchiare)</p> <p>Nome: <u>Utente</u> .....</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%;">attributo</th> <th style="width: 33%;">dominio</th> <th style="width: 33%;">moltepl. (*)</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p> <p><u>[V. saldo_positivo]</u></p> <p><math>\forall u, d, v [Utente(u) \wedge DataOra(d) \wedge</math>  <math>saldo(u, d, v)] \rightarrow v \geq 0</math></p> <p><u>[V. offre_scambo]</u></p> <p><math>\forall u, s, a [Utente(u) \wedge offre(s, u) \wedge</math>  <math>abi_sca(a, s)] \rightarrow fornisce(a, u)</math></p>	attributo	dominio	moltepl. (*)																															<p>2 Tipo: <u>Entità</u>   Relationship (cerchiare)</p> <p>Nome: <u>Utente</u> .....</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%;">attributo</th> <th style="width: 33%;">dominio</th> <th style="width: 33%;">moltepl. (*)</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p> <p><u>[V. no_auto_scambi]</u></p> <p><math>\forall u, s [Utente(u) \wedge Scambio(s) \wedge</math>  <math>offre(s, u)] \rightarrow \neg richiede(s, u)</math></p>	attributo	dominio	moltepl. (*)																														
attributo	dominio	moltepl. (*)																																																																	
attributo	dominio	moltepl. (*)																																																																	



3 Tipo: Entità | Relationship (cerchiare)Nome: Utente .....

attributo	dominio	moltepl. (*)

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

Vincoli:

[V.no-2-scambi-contemporaneamente] $\forall u, s1, s2, g1, g2, d1, d2$  $[Utente(u) \wedge s1 \neq s2 \wedge$  $\wedge offre(s1, u) \wedge offre(s2, u) \wedge giorno(s1, g1)$  $\wedge giorno(s2, g2) \wedge durata(u, s1, d1) \wedge$  $durata(s2, d2)] \rightarrow$  $[g1 + d1 < g2 \vee g2 + d2 < g1]$ 5 Tipo: Entità | Relationship (cerchiare)

Nome: .....

attributo	dominio	moltepl. (*)

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

Vincoli:

$$\left( ScambioAccettato(s1) \wedge \right. \\ \left. ScambioAccettato(s2) \right)$$
4 Tipo: Entità | Relationship (cerchiare)

Nome: .....

attributo	dominio	moltepl. (*)

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

Vincoli:

6 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

**moltiplicatore**( $a: Abilita, d: DataOra$ ): Reale  $\geq 0$

• pre-cond:  $Fornisce(a, this)$

• post-cond:  $S = \{ (s, f) \mid \text{offre}(this, s) \wedge \exists g \text{ giorno}(s, g) \wedge g < d \wedge \text{feedback}(s, f) \wedge \text{abi\_sca}(a, s) \}$

Result e' tale che:

$$|S| = 0 \rightarrow \text{Result} = 1 \wedge$$

$$|S| \neq 0 \rightarrow \text{Result} = 1 + \arctan\left(0.05 * \sum_{(s, f) \in S} f \cdot \frac{1}{|S|}\right)$$

**valore**( ): Reale  $\geq 0$

• pre-cond: nessuna

• post-cond:

Sia  $d$  tale che:  $\text{durata}(this, d)$

Sia  $g$  tale che:  $\text{giorno}(this, g)$

Sia  $a$  tale che:  $\text{abi\_sca}(a, this)$

Sia  $u$  tale che:  $\text{offre}(this, u)$

Sia  $m$  tale che:  $\text{moltiplicatore}(u, a, g, m)$

$$\text{Result} = d \cdot m$$

**saldo**(~~data~~  $d: DataOra$ ): Reale

• pre-cond: nessuna

• post-cond:  $R = \{ (s, v) \mid \text{richiede}(s, this) \wedge \exists g \text{ giorno}(s, g) \wedge \text{~~per~~} g \leq d \wedge \text{valore}(s, v) \wedge \text{ScambioAccettato}(s) \}$

$$O = \{ (s, v) \mid \text{offre}(s, this) \wedge \exists g \text{ giorno}(s, g) \wedge g \leq d \wedge \text{valore}(s, v) \wedge \text{ScambioAccettato}(s) \}$$

$$\text{Result} = \left( \sum_{(s, v) \in O} v \right) - \left( \sum_{(s, v) \in R} v \right) + \infty$$

$$\infty := \text{~~valore}(this, \infty)~~ \text{ bonus}(this, \infty)$$

↑ saldo al momento dell'iscrizione



## Tipi di Dato Reale

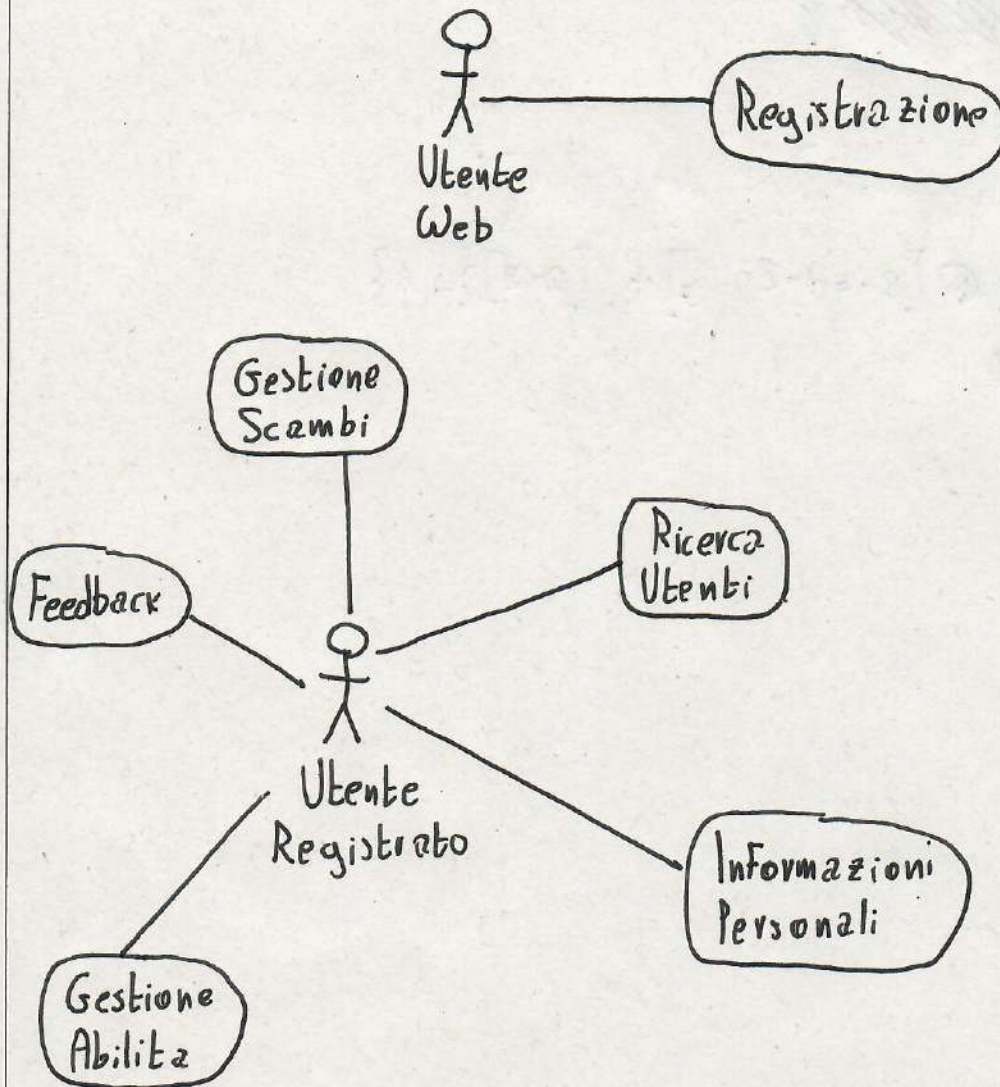
$$V_{020} = \text{Integer} \rightarrow \text{Integer}$$

CELL = [0-9]{12} // Prefixo {2} e Numero {10}

$$\text{Real\_TEN} = 1 \leq \text{Real} \leq 10$$

**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: Registrazione ..... (nome use-case)

Operazioni dello use-case:

registra(n:String, c:String, e:Mail, t:CELL):Utente

2 Specifica use-case: Gestione Abilita ..... (nome use-case)

Operazioni dello use-case:

aggiungi\_abilita(a:Abilita)

rimuovi\_abilita(a:Abilita)

3 Specifica use-case: Feedback ..... (nome use-case)

Operazioni dello use-case:

lascia-Feedback(s:ScambioAccettato, v:Voto):ScambioAccettato



4 Specifica use-case: *Gestione Scambi* ..... (nome use-case)

Operazioni dello use-case:

*accetta\_scambio(u:Scambio):ScambioAccettato*

*richiedi\_scambio(u:Utente, a:Abilita, g:DataOra, d:Real $\geq$ 0):Scambio*  
*desc:String a*

5 Specifica use-case: *Ricerca Utenti* ..... (nome use-case)

Operazioni dello use-case:

*ottieni\_abilita(u:Utente):Abilita[0..\*]*

*utenti-qualificati(a:Abilita, m:Real-TEN, d:DataOra):Utente[0..\*]*

6 Specifica use-case: *Informazioni Personali* ..... (nome use-case)

Operazioni dello use-case:

*calcola\_saldo(u:Utente, d:DataOra):Real-GE2*

*calcola\_molt(u:Utente, d:DataOra, a:Abilita):Real $\geq$ 0*

7 Specifica use-case: ..... (nome use-case)

Operazioni dello use-case:



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

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

Risposta

$\text{calcola\_molt} (u: \text{Utente}, d: \text{DataOra}, a: \text{Abilita}): \text{Real} \geq 0$

• pre-cond:  $\text{Fornisce}(a, u)$

• post-cond:  $\text{Result e' tale che} ::= \text{moltiplicatore}(u, a, d, \text{Result})$

$\text{Utenti\_qualificati} (a: \text{Abilita}, m: \text{Real\_TEN}, d: \text{DataOra}): \text{Utente} [0..*]$

• pre-cond: nessuna

• post-cond:  $U = \{ u \mid \text{Fornisce}(a, u) \wedge \exists k \text{ media\_utente}(u, a, d, k) \wedge k \geq m \}$

$\text{Result} = U$



Risposta alla Domanda 5 (segue)

$media\_utente(u: Utente, a: Abilita, d: DataOra): Real-TEN$

• pre-cond:  $fornisce(a, u)$

• post-cond:

$$S = \left\{ (s, f) \mid \begin{array}{l} offre(u, s) \wedge \exists g \text{ giorno}(s, g) \wedge g \leq d \\ \wedge \text{abi\_sca}(a, s) \wedge \text{Feedback}(s, f) \end{array} \right\}$$

Result e' tale che:

$$|S| = 0 \rightarrow \text{Result} = 1 \wedge$$

$$|S| \neq 0 \rightarrow \text{Result} = \sum_{(s, f) \in S} f \cdot \frac{1}{|S|}$$



## 2 Progettazione della base dati e delle funzionalità

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

- progettare una corrispondenza tra i domini concettuali ed opportuni domini SQL (domini base o utente, oppure realizzati mediante relazioni aggiuntive) supportati dal DBMS scelto
- eliminare attributi multivalore o composti
- eliminare relazioni is-a e generalizzazioni
- definire un identificatore primario per ogni entità
- valutare se e come aggiungere ridondanza in maniera controllata
- ristrutturare i vincoli esterni per renderli consistenti con la struttura del nuovo diagramma.

Descrivere brevemente le principali scelte effettuate.

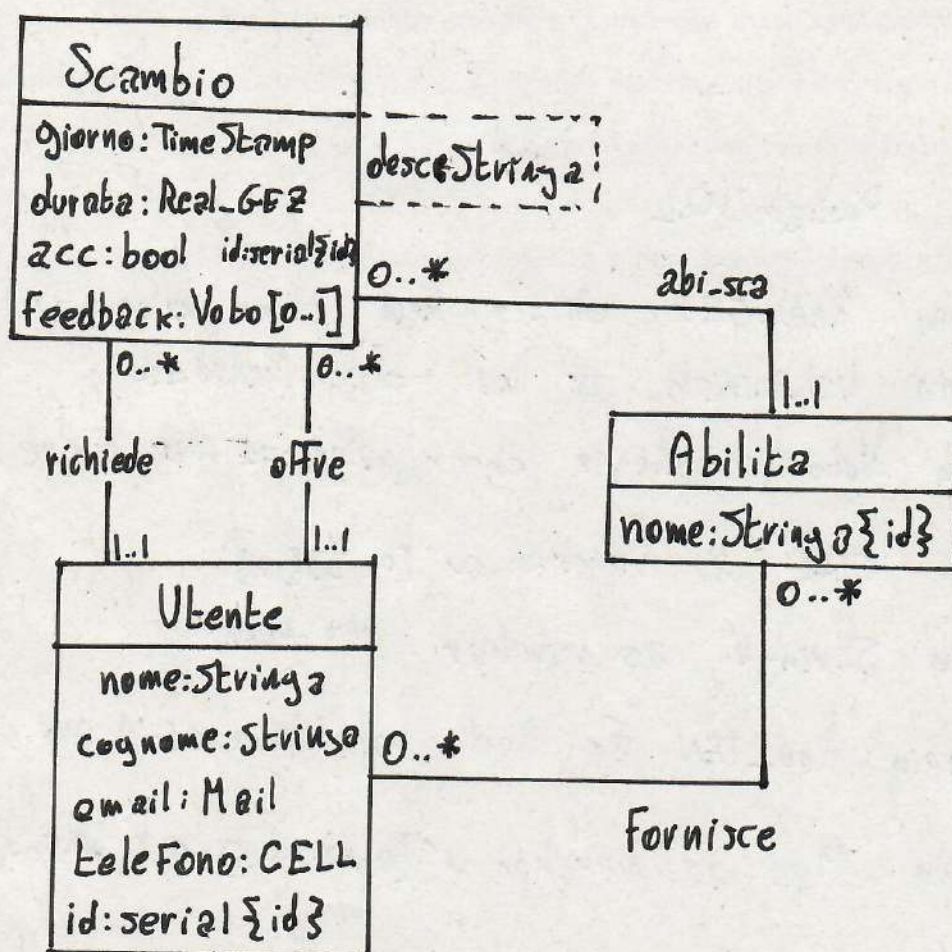
DBMS da utilizzare PostgreSQL

Corrispondenza tra domini concettuali e domini supportati dal DBMS

```
create domain Real-GEZ as Real check(value >= 0);
create domain Val-iniziale as Real check(value <= 0);
create domain Voto as Integer check(value >= 1 AND value <= 10);
create domain CELL as varchar ~ '[0-9]{12}'
create domain Stringa as varchar NOT NULL;
create domain Real-TEN as Real check(value >= 1 AND value <= 10);
create domain Mail as varchar ~ '[a-zA-Z0-9]+@[a-zA-Z0-9]+\.[a-z]{2,6}'
```



## Diagramma ER ristrutturato





Breve descrizione delle scelte effettuate durante la ristrutturazione

Fusione su Scambio

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)

[V. Feedback\_se - accettato]

$$\forall s [Scambio(s) \wedge \exists k feedback(s, k)] \rightarrow acc(s, True)$$

[V. no. 2. scambi contemporaneamente]

$$\forall u, s1, s2, g1, g2, d1, d2 [Ubente(u) \wedge s1 \neq s2 \wedge Scambio(s1) \wedge Scambio(s2) \\ \wedge acc(s1, True) \wedge acc(s2, True) \wedge offve(u, s1) \wedge offve(u, s2) \wedge giorno(s1, g1) \\ \wedge giorno(s2, g2) \wedge durata(s1, d1) \wedge durata(s2, d2)] \rightarrow$$

$$[g1 + d1 < g2 \vee g2 + d2 < g1]$$



**Domanda 7 (30 minuti; 60 minuti al massimo)** Proseguire la fase di progettazione logica della base di dati producendo lo schema relazionale della base dati e i relativi vincoli a partire dallo schema ER ristrutturato.

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

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

Attributi	<u>nome</u>	<u>cognome</u>	<u>email</u>	<u>telefono</u>	<u>id</u>	<u>bonus</u>		
Domini	<u>Stringa</u>	<u>Stringa</u>	<u>Mail</u>	<u>CELL</u>	<u>serial</u>	<u>val-intide</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 accorpa le relazioni che implementano le seguenti relationship: .....

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

Attributi	<u>nome</u>							
Domini	<u>Stringa</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 accorpa le relazioni che implementano le seguenti relationship: .....

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

Attributi	<u>utente</u>	<u>abilita</u>						
Domini	<u>Integer</u>	<u>Stringa</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):

foreign key abilita references Abilita(nome);

foreign key utente references Utente(id);

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

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

Attributi	<u>giorno</u>	<u>durata</u>	<u>id</u>	<u>acc</u>	<u>Feedback*</u>	<u>abilita</u>	<u>richiede</u>	<u>offre</u>
Domini	<u>Timestamp</u>	<u>Real-GE2</u>	<u>serial</u>	<u>bool</u>	<u>Voto</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): check(richiede < > offre);

foreign key abilita ref Abilita(nome); fk richiede ref Utente(id);

fk offre ref Utente(id);

La relazione accorpa le relazioni che implementano le seguenti relationship: richiede, offre, abi\_xo

5 Relazione Scambio ..... (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):

check(Feedback is NULL OR (Feedback is NOT NULL AND acc = True));

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



**Ulteriori vincoli esterni**

Per ogni ulteriore vincolo esterno (non ancora espresso perché non definibile mediante vincoli di chiave, foreign key, enunpla, dominio, inclusione), progettare un trigger che lo implementi, definendo: (a) gli eventi da intercettare (inserimento, modifica, eliminazione di enunpla); (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.no-2-scambi-contemporaneamente**

op: Insert o Update Scambio

```
Error = EXISTS (SELECT
                  FROM Scambio s WHERE new.id <> s.id AND s.utente
                  = new.utente AND
```

```
(s.giorno, s.giorno + cast(s.durata as 'hour')::Timestamp) OVERLAPS
(new.giorno, new.giorno + cast(new.durata as 'hour')::Timestamp));
```

```
if Error=True : errore e rollback
else : permetti oper.
```

**T.saldo-positivo**

op: Insert o Update Scambio

```
S = SELECT saldo(new.richiede, new.giorno) as sal;
```

```
if (S.sal < 0): Errore e rollback
```

```
else: permetti oper.
```

**T.offre-scambio**

op: Insert o Update Scambio

```
OK = EXISTS (SELECT * FROM fornisce f WHERE
              f.utente = new.offre AND f.abilita = new.abilita);
```

```
if OK=True : permetti op.
```

```
else: errore e rollback
```



**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

*media*

Create Function ~~media~~ (*u*:Integer, *a*:Stringa, *d*:TimeStamp):<sup>Real-TEN</sup>~~Real-TEN~~

E=EXISTS(SELECT \* FROM Fornisce WHERE *abilita*=*a* AND  
*utente*=*u*);

if (E==False): termina operazione

Q=SELECT ~~media~~ AVG(Feedback) as *m*, count(\*) as *n*  
FROM Scambio WHERE *offre*=*u* AND *giorno*<*d* AND *acc*=True  
AND *abilita*=*a* AND Feedback IS NOT NULL;

if (Q.n==0): Result=1

else: Result = Q.m

Create Function *valore* (*s*:Integer):Real-GEZ

Q=SELECT *durata*\**moltiplicatore*(*offre*, *abilita*, *giorno*) as *val*  
FROM Scambio SC WHERE *id*=*s*;

Result = Q.val

Create Function *moltiplicatore* (*u*:Integer, *a*:Stringa, *d*:TimeStamp):Real-GEZ

Q=SELECT 1+ARCTAN(0.05\**media*(*u*, *a*, *d*)) as *mol*;

Result = Q.mol



Risposta alla Domanda 8 (segue) Timestamp

Saldo( $u$ :Integer,  $d$ :~~DateTime~~):Real

Q=WITH R as (SELECT sum(valore(id)) as ric  
FROM Scambio WHERE richiede= $u$  AND  
giorno  $\leq$   $d$  AND acc=True)

O as (SELECT sum(valore(id)) as off  
FROM Scambio WHERE offre= $u$  AND  
giorno  $\leq$   $d$  AND acc=True)

SELECT bonus+Q.off-R.ric as saldo  
FROM Utente, R, Q  
WHERE id= $u$

Result = Q.saldo

Utenti-qualificati( $a$ :Stringa,  $m$ :Real-TEN,  $d$ :Timestamp):Insieme( $\langle$ Integer $\rangle$ )

Q= SELECT  $u$ .id  
FROM Utente  $u$  JOIN Fornisce F on  $u$ .id=F.utente  
WHERE F.abilita= $a$  AND  
media( $u$ .id,  $a$ ,  $d$ )  $\geq$   $m$ ;

Result = Q