

Esame Es.20230317 – Prova scritta del 17 Marzo 2023

Si richiede di progettare *AllenaMondo*, un sistema web che realizza una rete sociale per la gestione delle attività sportive personali e collettive, amatoriali e competitive. A tal fine si trascrive di seguito il risultato della raccolta dei requisiti con alcuni esperti del dominio.

Gli utenti di *AllenaMondo* sono gli sportivi. Uno **sportivo** (o utente) è identificato dal proprio **nome utente** (una stringa). Inoltre, di uno sportivo è di interesse conoscere **nome, cognome, il genere** (maschio o femmina) e la sua **data di nascita**. Opzionalmente, un utente può anche fornire al sistema il proprio **peso e la propria altezza**. La socialità del sistema *AllenaMondo* si realizza nella **possibilità per due sportivi di essere amici**. Un utente può avere molti amici.

Gli utenti usano *AllenaMondo* per tracciare la loro attività sportiva. Di ogni **sport** memorizzato nel sistema interessa il **nome** ed una **descrizione testuale**.

Gli sportivi interagiscono con *AllenaMondo* comunicando le loro attività di allenamento quotidiane (via web o app per smartphone). In particolare, di ogni **attività di allenamento** di un utente, *AllenaMondo* memorizza lo **sport esercitato**, gli **istanti di inizio e di fine**, e (opzionalmente) un **valore tra 1 e 5 che rappresenta la "difficoltà"** dell'allenamento, come percepita dallo stesso utente. Inoltre, durante un allenamento, uno sportivo può chiedere a *AllenaMondo* di memorizzare il **percorso seguito**, usando il suo smartphone. In questo caso, l'app per smartphone di *AllenaMondo*, durante l'allenamento, comunica periodicamente (ogni qualche secondo) al sistema le **coordinate geografiche (latitudine e longitudine)** della posizione dell'utente. Il percorso rappresentato dalla sequenza di tali coordinate (con i relativi **istanti di tempo**) viene associato all'attività di allenamento. Successivamente, l'utente può decidere di rendere **pubblico tale percorso** (assegnandogli un **nome** ed un **commento testuale**), di modo che anche gli altri utenti possano visualizzarlo su una mappa.

Una caratteristica innovativa del sistema *AllenaMondo* è la possibilità per gli sportivi di sfidarsi, singolarmente oppure in squadre. In particolare, ogni utente può creare una **nuova sfida**, assegnandole una tipologia ("**tra squadre**" o "**tra singoli**"), **un nome, il periodo di durata, la durata minima di un allenamento affinché sia valido per la sfida, gli sport esercitabili (uno o più)** e la categoria ("**più chilometri**" o "**più allenamenti**"). In caso una sfida sia tra squadre, l'utente creatore della sfida deve anche definire il **numero massimo di partecipanti per ogni squadra**.

Una volta che una sfida viene creata, ogni utente può parteciparvi, singolarmente (se la sfida è "**tra singoli**") oppure iscrivendosi ad una squadra (se la sfida è "**tra squadre**"). Delle **squadre**, che sono **limitate ad una singola sfida**, interessa il **nome** e l'**insieme dei componenti**. Si noti che uno sportivo non può appartenere a due squadre nell'ambito di una stessa sfida.

La partecipazione da parte di uno sportivo ad una sfida si sostanzia nell'effettuare attività di allenamento relative agli sport e nell'arco temporale previsti dalla sfida. La sfida è vinta dall'individuo o squadra che totalizza il massimo punteggio. Per le sfide di categoria "**più chilometri**", il punteggio di un utente o squadra è la somma dei chilometri percorsi nelle sue attività di allenamento validi per la sfida. Per le sfide di categoria "**più allenamenti**", il punteggio è invece dato dal numero complessivo di allenamenti validi. *AllenaMondo* deve associare ad ogni utente tutte le sfide da lui create o a cui ha partecipato.

Il sistema *AllenaMondo* deve fornire le seguenti funzionalità.

- Registrazione al servizio da parte di un utente web.

- Inserimento di un nuovo sport da parte di un amministratore del sistema. ✓
- Inserimento di una nuova attività di allenamento da parte di uno sportivo, con tutte le informazioni richieste. ✓
- Creazione di una sfida da parte di uno sportivo, con tutte le caratteristiche necessarie. ✓
- Creazione di una squadra per una sfida da parte di uno sportivo, con tutte le caratteristiche necessarie. ✓
- Sottoscrizione di uno sportivo a una sfida (come singolo o come membro di una squadra). ✓

Ogni utente deve poter calcolare un resoconto degli allenamenti che ha effettuato in un dato periodo di tempo. In particolare, per ognuno degli sport praticati, si vuole conoscere il tempo totale di allenamento ed il numero totale di chilometri percorsi.

Infine, il sistema deve permettere il calcolo dello sport maggiormente praticato dagli amici di un certo sportivo in un dato periodo di tempo. In particolare, lo sport maggiormente praticato è quello a cui è stato dedicato il maggior tempo totale (si noti che potrebbero esserci più sport nel risultato, in caso il tempo totale praticato per questi dovesse essere lo stesso).

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 :

- Username: Stringa
- nome: Stringa
- cognome: Stringa
- genere: Generale
- dataNascita: Data
- pesoKg: Reale > 0
- altezzaCm: Intero > 0
- associato ai suoi Utenti amici
- associato agli Sport che pratica

Sport:

- nome: Stringa
- desc: Stringa
- associato alle Attività svolte

Attività:

- inizioAtt: DataOra
- fineAtt: DataOra
- diff: 1..5

Posizione:

- longitudine: Reale
- latitudine: Reale
- istante: DataOra
- Specializzazione:

 - Pubblico:

 - nome: Stringa
 - conn: Stringa
 - associato all' Attività

Squadra:

- nome: Stringa
- associato agli Utenti partecipanti

Sfido:

- nome: Stringa
- inizioSf: DataOra
- fine Sf: DataOra
- associato agli Sport svolti
- numero Sfido:

 - Squadre:

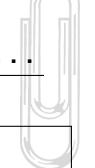
 - nmax: Interosso
 - associato alle Squadre

 - Singolo:

 - associato agli Utenti

Tipo Sfido:

- tempo
- Allenamenti



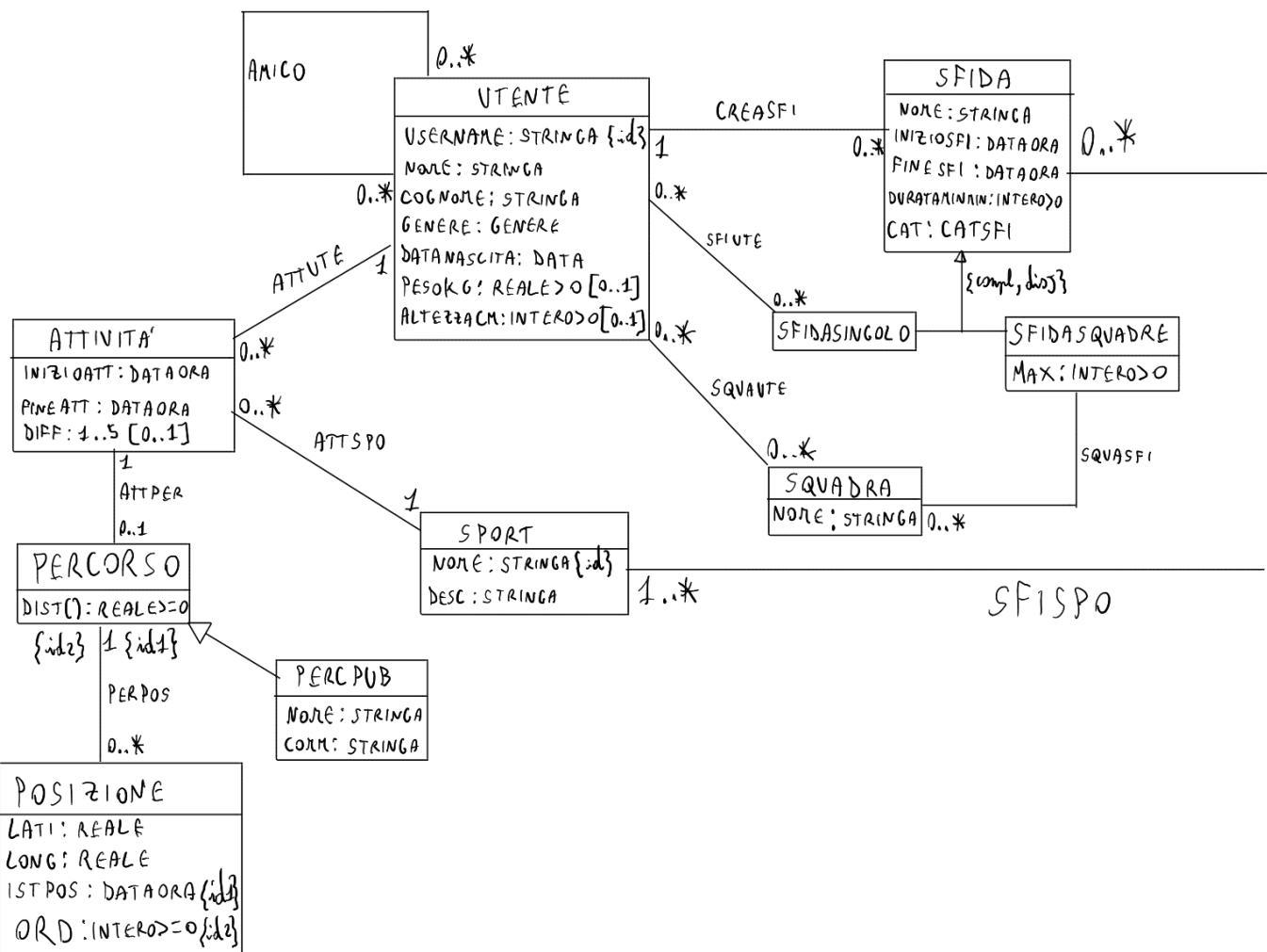
Risposta alla Domanda 1 (segue)

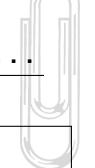
Domanda 2 (45 minuti; 75 minuti al massimo) Proseguire la fase di Analisi Concettuale dei requisiti, producendo un diagramma ER concettuale per l'applicazione, il dizionario dei dati ed eventuali vincoli esterni.

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.





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>V.TENTE</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;">moltep. (*)</th></tr> </thead> <tbody> <tr> <td></td><td></td><td></td></tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli: $\forall .V.TENTE . NOAUTOAMICIZIA$ $\forall u, u'. AMICO(u, u') \rightarrow u \neq u'$ $\forall .V.TENTE . NO2SQUADRE$ $\forall u . VENTE(u) \rightarrow \exists s, s', m \quad s \neq s' \wedge$ $SQUAUTE(u, s) \wedge SQUAUTE(u, s') \wedge SQUASFI(s, m) \wedge SQUASFI(s', m)$ $\forall .V.TENTE . SFIDADOPONASCITA$ $\forall u, s, i, dn . VENTE(u) \wedge SFIDA(s) \wedge INIZIOSFI(s, i) \wedge DATANASC(u, dn)$ $\wedge [CREASFI(u, s) \vee SFIVTE(u, s) \vee (\exists sg . SQUAUTE(sg, u) \wedge SQUASFI(sg, m))]$ $\wedge dn . DATA(dn, i) \rightarrow dn \geq dn$ </p>	attributo	dominio	moltep. (*)				<p>[2] Tipo: Entità Relationship (cerchiare)</p> <p>Nome: ...<u>S.QVADRA</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;">moltep. (*)</th></tr> </thead> <tbody> <tr> <td></td><td></td><td></td></tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli: $\forall .V.SQVADRA . MAX VENTI$ $\forall sg, sf, m . SQUASFI(sg, sf) \wedge MAX(sf, m)$ $\left \{ u SQUAUTE(u, sg) \} \right \leq m$ </p>	attributo	dominio	moltep. (*)			
attributo	dominio	moltep. (*)											
attributo	dominio	moltep. (*)											

3 Tipo: Entità Relationship (cerchiare)		
Nome: ...SFIDA.....		
attributo	dominio	moltep. (*)

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

Vincoli:

$$\begin{aligned} & \forall s, i, f \text{ SFIDA}(s) \wedge \text{INIZIOSFI}(i) \wedge \\ & \text{FINESSI}(f) \rightarrow i \leq f \end{aligned}$$

5 Tipo: Entità Relationship (cerchiare)		
Nome: ...ATTIVITA'		
attributo	dominio	moltep. (*)

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

Vincoli:

$$\begin{aligned} & \forall a, i, f \text{ ATTIVITA'}(a) \wedge \text{INIZIATT}(i) \wedge \text{FINEATT}(f) \\ & \rightarrow i \leq f \\ & \forall a, u, i, d, di \text{ ATTUTE}(a, u) \wedge \text{INIZIOATT}(a, i) \wedge \\ & \text{DATANASC}(d, u) \wedge \text{DATA}(di, i) \rightarrow d \leq di \end{aligned}$$

4 Tipo: Entità Relationship (cerchiare)		
Nome: ...POSIZIONE.....		
attributo	dominio	moltep. (*)

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

Vincoli:

$$\begin{aligned} & \forall p, o \text{ POSIZIONE}(p) \wedge \text{ORD}(p, o) \wedge o \geq 1 \wedge \text{PERPOS}(p, p_o) \rightarrow \\ & \exists p' \text{ POSIZIONE}(p') \wedge \text{ORD}(p', o-1) \wedge \text{PERPOS}(p, p_{o'}) \\ & \forall . POSIZIONE. ORDINEORA \\ & \forall p, o, i, p', o', i' \text{ POSIZIONE}(p) \wedge \text{POSIZIONE}(p') \wedge \text{ORD}(p, o) \\ & \wedge \text{ORD}(p', o') \wedge \text{ISTPOS}(p, i) \wedge \text{PERPOS}(p, p_i) \wedge \text{PERPOS}(p', p_{i'}) \wedge \\ & \text{PERPOS}(p', p_{i'}) \wedge \text{ATTPER}(att, p_{i'}) \wedge \text{INIZIOATT}(i, att) \wedge \\ & \text{FINEATT}(f, att) \rightarrow i \leq i' \wedge i' \leq f \\ & \forall p, o, i, p', o', i' \text{ POSIZIONE}(p) \wedge \text{POSIZIONE}(p') \wedge \text{ORD}(p, o) \\ & \wedge \text{ORD}(p', o') \wedge \text{ISTPOS}(p, i) \wedge \text{ISTPOS}(p', i') \wedge \text{PERPOS}(p, p_i) \wedge \text{PERPOS}(p', p_{i'}) \\ & \wedge o > o' \rightarrow i > i' \end{aligned}$$

6 Tipo: Entità Relationship (cerchiare)		
Nome:		
attributo	dominio	moltep. (*)

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

Vincoli:

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

Nome:

attributo	dominio	moltepl. (*)

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

Vincoli:

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

Nome:

attributo	dominio	moltepl. (*)

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

Vincoli:

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

Nome:

attributo	dominio	moltepl. (*)

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

Vincoli:

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

Nome:

attributo	dominio	moltepl. (*)

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

Vincoli:

<p>11 Tipo: Entità Relationship (cerchiare)</p> <p>Nome:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">attributo</th> <th style="text-align: left; padding: 2px;">dominio</th> <th style="text-align: left; padding: 2px;">moltepl. (*)</th> </tr> </thead> <tbody> <tr> <td colspan="3" style="height: 100px;"></td> </tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p>	attributo	dominio	moltepl. (*)				<p>13 Tipo: Entità Relationship (cerchiare)</p> <p>Nome:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">attributo</th> <th style="text-align: left; padding: 2px;">dominio</th> <th style="text-align: left; padding: 2px;">moltepl. (*)</th> </tr> </thead> <tbody> <tr> <td colspan="3" style="height: 100px;"></td> </tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p>	attributo	dominio	moltepl. (*)			
attributo	dominio	moltepl. (*)											
attributo	dominio	moltepl. (*)											

<p>12 Tipo: Entità Relationship (cerchiare)</p> <p>Nome:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">attributo</th> <th style="text-align: left; padding: 2px;">dominio</th> <th style="text-align: left; padding: 2px;">moltepl. (*)</th> </tr> </thead> <tbody> <tr> <td colspan="3" style="height: 100px;"></td> </tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p>	attributo	dominio	moltepl. (*)				<p>14 Tipo: Entità Relationship (cerchiare)</p> <p>Nome:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">attributo</th> <th style="text-align: left; padding: 2px;">dominio</th> <th style="text-align: left; padding: 2px;">moltepl. (*)</th> </tr> </thead> <tbody> <tr> <td colspan="3" style="height: 100px;"></td> </tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p>	attributo	dominio	moltepl. (*)			
attributo	dominio	moltepl. (*)											
attributo	dominio	moltepl. (*)											

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

GENERE: ('M', 'F')

CATSEI: ('ALL', 'KM')

DURATA: (ORE=INTERO ≥ 0 , MINUTI $\in 0..60$, SECONDI $\in 0..60$)

PERCORSO.DIST(): REALE ≥ 0

- pre:

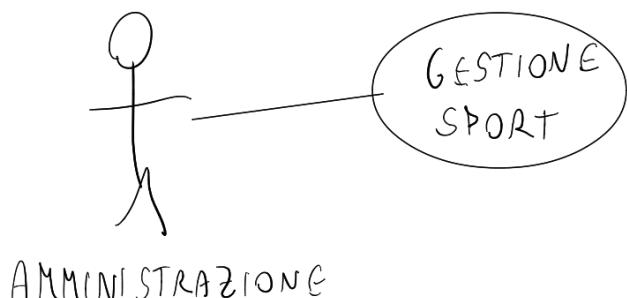
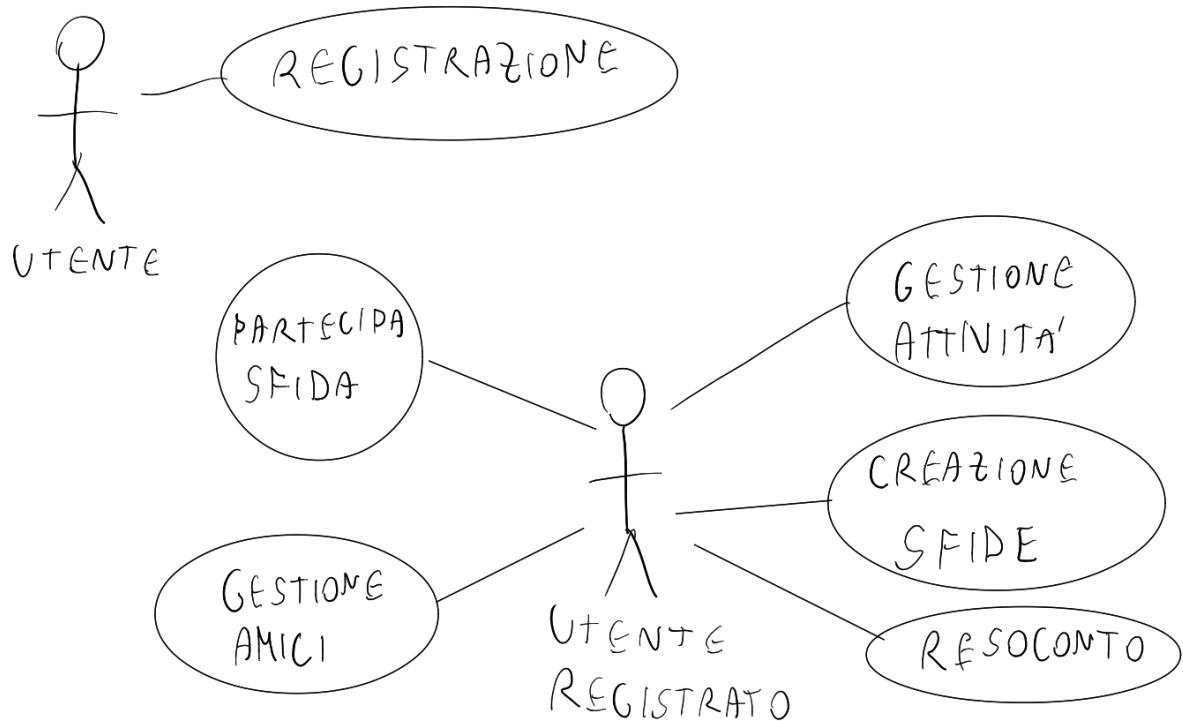
- post:

$$P = \left\{ (p_1, p_2, d) \mid \begin{array}{l} \text{PERPOS(this, } p_1) \wedge \text{PERPOS(this, } p_2) \wedge [\text{lat}, \text{lat}', \text{lon}, \text{lon}', \sigma, \sigma' \\ \wedge \text{LAT(lat, } p_1) \wedge \text{LAT(lat', } p_2) \wedge \text{CON(lon, } p_1) \wedge \text{CON(lon', } p_2) \wedge \text{ORD(p}_1, \sigma) \wedge \text{ORD(p}_2, \sigma') \\ \wedge d = \text{DISTANZA(lat, lat', lon, lon')} \wedge \sigma' = \sigma + 1 \end{array} \right\}$$

$$\text{RETURN} = \text{SUM}(d) \\ (p_1, p_2, d) \in P$$

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

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: REGISTRATION (nome use-case)

Operazioni dello use-case:

NEWUTENTE($u: \text{STRINGA}, n: \text{STRINGA}, c: \text{STRINGA}, dn: \text{DATA}, g: \text{GENERE}, h: \text{REAL} \in [0..1], a: \text{INTERO} \in [0..1]\rangle$): UTENTE

2 Specifica use-case: GESTIONE ATTIVITA' (nome use-case)

Operazioni dello use-case:

NEWATTIVITA'($i: \text{DATAORA}, f: \text{DATAORA}, d: 1..5 \in [0..1], s: \text{SPORT}$): ATTIVITA'

NEWPERCORSO($a: \text{ATTIVITA'}$): PERCORSO

NEWPERCPUB($n: \text{STRINGA}, c: \text{STRINGA}, p: \text{PERCORSO}$): PERCPUB

NEWPOSIZIONE($lat: \text{REAL}, lon: \text{REAL}, i: \text{DATAORA}, o: \text{INTERO}, p: \text{PERCORSO}$): POSIZIONE

3 Specifica use-case: CREAZIONE SFIDE (nome use-case)

Operazioni dello use-case:

CREASFIDASINGOLA($n: \text{STRINGA}, i: \text{DATAORA}, f: \text{DATAORA}, d: \text{INTERO}, c: \text{CATSFI}, s: \text{SPORT} [1..*]$): SFIDASINGOLA

CREASFIDASQUADRE($n: \text{STRINGA}, i: \text{DATAORA}, f: \text{DATAORA}, d: \text{INTERO}, c: \text{CATSFI}, s: \text{SPORT} [1..*], m: \text{INTERO}$): SFIDASQUADRE

4 Specifica use-case: PARTECIPA SFIDE (nome use-case)

Operazioni dello use-case:

PARTECIPASINGOLO(λ ; SFIDASINGOLO)

CREASQUADRA(n ; STRINGA, λ ; SFIDASQUADRA); SQUADRA

PARTECIPASQUADRA(λ ; SFIDASQUADRA)

WINSFIDASQUADRA(λ ; SFIDASQUADRA): SQUADRA[0..*]

WINSFIDASINGOLO(λ ; SFIDASINGOLO): UTENTE[0..*]

5 Specifica use-case: GESTIONE AMICI (nome use-case)

Operazioni dello use-case:

NEWAMICO(u ; UTENTE)

MAXSPORTAMICO(u ; UTENTE, u' ; UTENTE): SPORT[0..*]

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

Operazioni dello use-case:

RESOCONTOSPORT(u ; UTENTE, i ; DATAORA, f ; DATAORA):

(SPORT, DURATA, REALE ≥ 0)[0..*]

7 Specifica use-case: GESTIONE SPORT (nome use-case)

Operazioni dello use-case:

NEWSPORT(n ; STRINGA, d ; STRINGA): SPORT

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

$\text{MAXSPORTAMICO}(\text{u}; \text{UTENTE}, \text{u}'; \text{UTENTE}, \text{i}; \text{DATAORA}, \text{f}; \text{DATAORA}) ; \text{SPORT}[0..*]$

- pre: $\text{AMICO}(\text{u}, \text{u}') \wedge \text{i} \leq \text{f}$

- post:

$$S = \left\{ (\text{s}, \text{t}) \mid \begin{array}{l} \exists \alpha, \text{ai}, \text{af} \text{ ATTUTE}(\alpha, \text{u}') \wedge \text{ATTSPo}(\alpha, \text{s}) \wedge \\ \text{INIZIOATT}(\text{i}, \alpha) \wedge \text{FINEATT}(\text{af}, \alpha) \wedge \text{ai} \geq \text{i} \wedge \text{af} \leq \text{f} \wedge \\ T = \{(\alpha, \text{af}) \mid \text{ATTSPo}(\alpha, \text{s}) \wedge \exists \text{i}, \text{f} \text{ INIZIOATT}(\text{i}, \alpha) \\ \wedge \text{FINEATT}(\text{f}, \alpha) \wedge \text{af} = \text{i} - \text{f} \wedge \text{ATTUTE}(\alpha, \text{u})\} \end{array} \right\}$$

$$\wedge t = \sum_{(\alpha, \text{af}) \in T} (\text{af})$$

$$\text{RESULT} = \left\{ \text{s} \mid \exists \text{t} \quad (\text{s}, \text{t}) \in \text{ARGMAX}(\text{t}) \right\}_{(\text{s}, \text{t}) \in S}$$

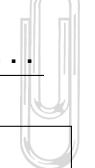
$\text{RESOCONTOSPORT}(\text{u}; \text{UTENTE}, \text{i}; \text{DATAORA}, \text{f}; \text{DATAORA}) :$

$(\text{SPORT}, \text{DURATA}, \text{REAL} \geq 0)[0..*]$

- pre: $\text{i} \leq \text{f} \wedge \exists \alpha \text{ ATTUTE}(\alpha, \text{u})$

- post:

$$\text{RESULT} = \left\{ (\text{s}, \text{d}, \text{t}) \mid \begin{array}{l} \exists \alpha, \text{ai}, \text{af} \text{ ATTUTE}(\alpha, \text{u}) \wedge \text{ATTSPo}(\alpha, \text{s}) \wedge \text{INIZIOATT}(\text{i}, \alpha) \\ \wedge \text{FINEATT}(\text{af}, \alpha) \wedge \text{ai} \geq \text{i} \wedge \text{af} \leq \text{f} \wedge \\ D = \{(\alpha, \text{ad}) \mid \text{ATTSPo}(\alpha, \text{s}) \wedge \text{ATTUTE}(\alpha, \text{u}) \wedge \exists \text{p} \text{ ATTPER}(\alpha, \text{p}) \wedge \text{DIST}(\text{p}, \text{ad})\} \wedge d = \sum_{(\alpha, \text{ad}) \in D} (\text{ad}) \wedge \\ T = \{(\alpha, \text{af}) \mid \text{ATTSPo}(\alpha, \text{s}) \wedge \exists \text{i}, \text{f} \text{ INIZIOATT}(\text{i}, \alpha) \\ \wedge \text{FINEATT}(\text{f}, \alpha) \wedge \text{af} = \text{i} - \text{f} \wedge \text{ATTUTE}(\alpha, \text{u})\} \wedge t = \sum_{(\alpha, \text{af}) \in T} (\text{af}) \end{array} \right\}$$



Risposta alla Domanda 5 (segue)

2 Progettazione della base dati e delle funzionalità

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

- progettare una corrispondenza tra i domini concettuali ed opportuni domini SQL (domini base o utente, oppure realizzati mediante relazioni aggiuntive) supportati dal DBMS scelto
- eliminare attributi 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

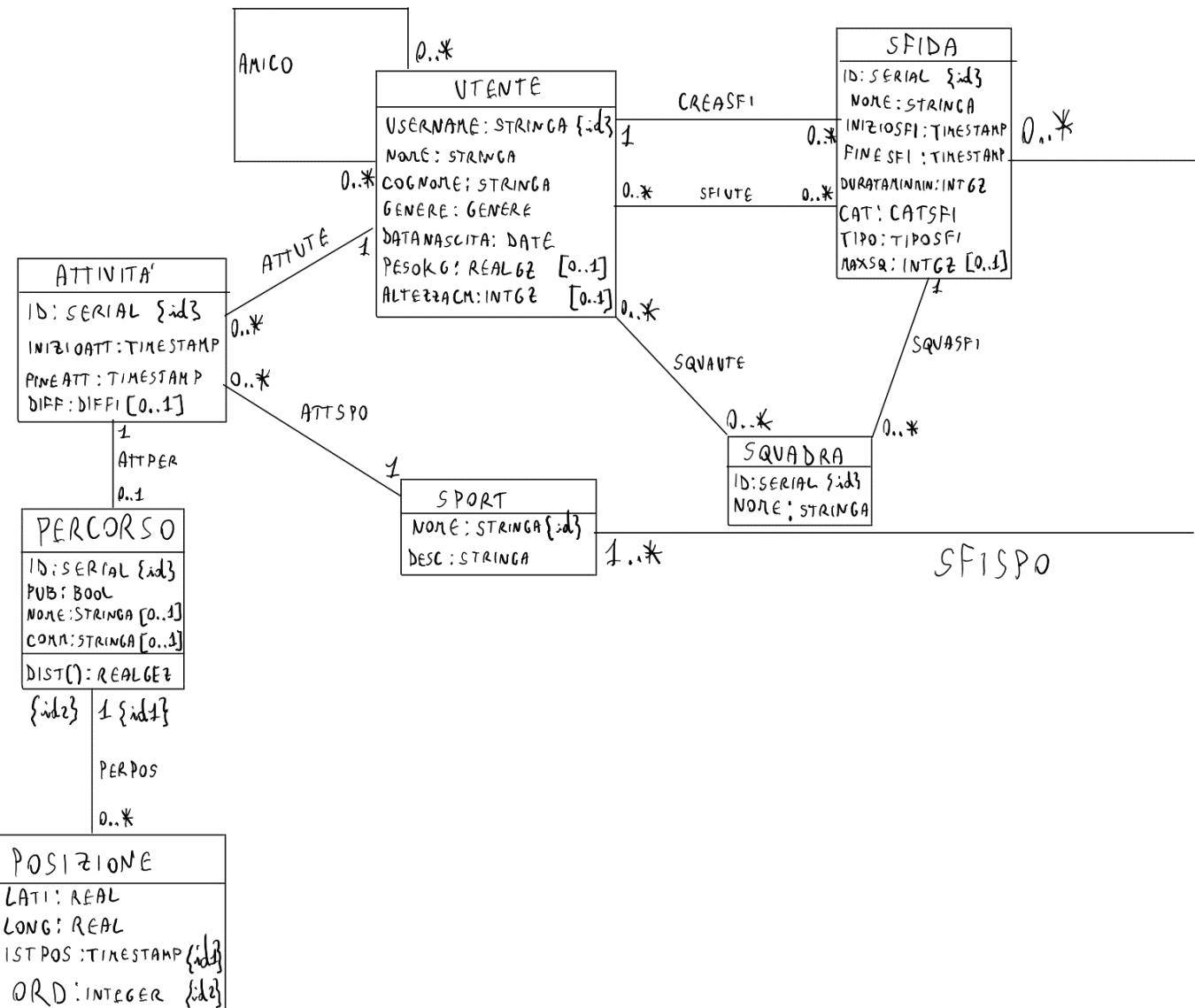
Corrispondenza tra domini concettuali e domini supportati dal DBMS

```

CREATE DOMAIN STRINGA AS VARCHAR NOT NULL;
CREATE DOMAIN INTGZ AS INTEGER CHECK(VALUE>0);
CREATE DOMAIN INTGEZ AS INTEGER CHECK(VALUE>=0);
CREATE DOMAIN REALGEZ AS REAL CHECK(VALUE>=0);
CREATE DOMAIN REALGZ AS REAL CHECK(VALUE>0);
CREATE DOMAIN DIFFI AS INTEGER CHECK(VALUE>=1 AND VALUE<=5);
CREATE TYPE CATSF1 AS ENUM ('ALL','KM');
CREATE TYPE TIPOSFI AS ENUM ('SING','SQUA');
CREATE TYPE GENERE AS ENUM ('M','F');

```

Diagramma ER ristrutturato



Breve descrizione delle scelte effettuate durante la ristrutturazione

Vincoli esterni introdotti o modificati durante la fase di ristrutturazione
 (si omettano i vincoli esterni la cui formulazione è rimasta identica a seguito della ristrutturazione)

$\forall s. SFIDA. SINGOLO$

$\forall u, s. SFIVTE(u, s) \rightarrow TIPOSFI(s, 'SING')$

$\forall s. SFIDA. SQUADRE$

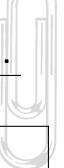
$\forall sq, sf. SQUASFI(sq, sf) \rightarrow TIPOSFI(sf, 'SQUA')$

$\forall s. SFIDA. MAXSQUADRA$

$\forall s. SFIDA(s) \wedge TIPOSFI(s, 'SQUA') \leftrightarrow \exists m. MAXSQ(s, m)$

$\forall p. PERCORSO(p)$

$\forall p. PERCORSO(p) \wedge PUB(p, 'TRUE') \leftrightarrow \exists n, c. NORE(p, n) \wedge COMM(p, c)$



Risposta alla Domanda 6 (segue)

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

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

1 Relazione ...UTENTE.... (nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>USERNAME</u> NOME COGNOME DATANASC GENERE PESO KG * ALTEZZA CM *	
Domini STRINGA STRINGA STRINGA DATE GENERE REALCE INT 67	

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:

2 Relazione ...AMICO.... (nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>UTENTE1</u> <u>UTENTE2</u>	
Domini STRINGA 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 (UTENTE1) REF UTENTE(USERNAME) CHECK(UTENTE1 <> UTENTE2)
FK (UTENTE2) REF UTENTE(USERNAME)

La relazione accorda le relazioni che implementano le seguenti relationship:

3 Relazione ...SPORT.... (nome)	Derivante da: entità relationship (cerchiare)
Attributi NOME DESC	
Domini STRINGA 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 accorda le relazioni che implementano le seguenti relationship:

4 Relazione ...ATTIVITA'.... (nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>ID</u> INIZIO ATT FINE ATT DIFF * UTENTE SPORT	
Domini SERIAL TIMESTAMP TIMESTAMP DIFF STRINGA 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) REF UTENTE(USERNAME) CHECK(INIZIO ATT < FINE ATT)
FK (SPORT) REF SPORT(NOME)

La relazione accorda le relazioni che implementano le seguenti relationship: ATTRUTTO, ATTSPO

5 Relazione ...PER CORSO... (nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>ID</u> PUB NOME * COMM * ATT	
Domini SERIAL BOOL STRINGA STRINGA INTEGER	

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

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

FK (ATT) REF ATTIVITA'(ID) UNIQUE(ATT)
CHECK((PUB=TRUE AND NOME<>NULL AND COMM<>NULL) OR (PUB=FALSE AND NOME=NULL AND COMM=NULL))

La relazione accorda le relazioni che implementano le seguenti relationship: ATT PER

6 Relazione	<u>POSIZIONE</u>	(nome)	Derivante da:	entità	relationship (cerchiare)
Attributi	<u>LAT</u>	<u>LONG</u>	<u>ISTPOS</u>	<u>ORD</u>	<u>PERC</u>
Domini	REAL	REAL	TIMESTAMP	INTEGER	INTEGER

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

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

FK(PERC) REF PERCORSO(ID)

UNIQUE (ISTPOS, PERC)

La relazione accorda le relazioni che implementano le seguenti relationship: PERPOS.....

7 Relazione	<u>S.FIDA</u>	(nome)	+	Derivante da:	entità	relationship (cerchiare)
Attributi	<u>ID</u>	<u>NOME</u>	<u>INIZIOSFI</u>	<u>FINESFI</u>	<u>DURMININ</u>	CAT
Domini	SERIAL	STRINGA	TIMESTAMP	TIMESTAMP	INTEGER	CATSF1

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(CREA) REF UTENTE(USERNAME)

CHECK(INIZIOSFI < FINESFI)

CHECK((TIPO='SQUA') = (MAXSQUA IS NOT NULL))

La relazione accorda le relazioni che implementano le seguenti relationship: CREASF1.....

8 Relazione	<u>S.FIDA</u>	(nome)	2	Derivante da:	entità	relationship (cerchiare)
Attributi	<u>CREA</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 accorda le relazioni che implementano le seguenti relationship:

9 Relazione	<u>S.FISPO</u>	(nome)	Derivante da:	entità	relationship (cerchiare)
Attributi	<u>SFIDA</u>	<u>SPORT</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(SFIDA) REF SFIDA(ID)

FK(SPORT) REF SPORT(NOME)

La relazione accorda le relazioni che implementano le seguenti relationship:

10 Relazione	<u>SQUADRA</u>	(nome)	Derivante da:	entità	relationship (cerchiare)
Attributi	<u>ID</u>	<u>NOME</u>	<u>S.FIDA</u>		
Domini	SERIAL	STRINGA	INTEGER		

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

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

FK(SFIDA) REF SFIDA(ID)

La relazione accorda le relazioni che implementano le seguenti relationship: SFISQVA.....

11 Relazione S.QVA.U.T.E... (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>UTENTE</u>	<u>SQUADRA</u>						
Domini	STRINGA	INTEGER						

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

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

FK(UTENTE) REF UTENTE(USERNAME)

FK(SQUADRA) REF SQUADRA(ID)

La relazione accorda le relazioni che implementano le seguenti relationship:

12 Relazione S.FI.V.T.E..... (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>UTENTE</u>	<u>SFIDA</u>						
Domini	STRINGA	INTEGER						

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

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

FK(UTENTE) REF UTENTE(USERNAME)

FK(SFIDA) REF SFIDA(ID)

La relazione accorda le relazioni che implementano le seguenti relationship:

13 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:

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.

T. SFIDA DOPO NASCITA CREA

- INSERT OR UPDATE SFIDA

- OK = EXIST (SELECT * FROM UTENTE

WHERE USERNAME = NEW.CREA AND DATA_NASC < CAST(NEW.INIZIOSFI AS DATE))

T. SFIDA DOPO NASCITA PART

- INSERT OR UPDATE

- ERROR = EXIST (SELECT * FROM UTENTE u, SFIDA s

WHERE u.USERNAME = NEW.UTENTE AND s.ID = NEW.SFIDA

AND u.DATA_NASC > CAST(s.INIZIOSFI AS DATE))

T. SFIDA DOPO NASCITA SQUA

- INSERT OR UPDATE SQUAUTE

- ERROR = EXIST (SELECT * FROM UTENTE u, SQUADRA sq, SFIDA sf

WHERE u.USERNAME = NEW.UTENTE AND sq.ID = NEW.SQUADRA AND sq.SFIDA = sf.ID

AND u.DATA_NASC > CAST(sf.INIZIOSFI AS DATE))

T. UTENTE, NO2 SQUADRE

- INSERT OR UPDATE SQUAUTE

- ERROR = EXIST (SELECT * FROM SQUAUTE su, SQUADRA sq

WHERE su.UTENTE = NEW.UTENTE AND su.SQUADRA = sq.ID AND sq.ID <> NEW.SQUADRA

AND sq.SFIDA = (SELECT SFIDA FROM SQUADRA
WHERE ID = NEW.SQUADRA)

T. SFIDA, MAX SQUADRA

- INSERT OR UPDATE SQUAUTE

- OK = (SELECT COUNT(UTENTE) FROM SQUAUTE

WHERE ID = NEW.SQUADRA) <= (SELECT MAXSQ FROM SFIDA sf, SQUADRA sq

WHERE sf.ID = sq.SFIDA AND sq.ID = NEW.SQUADRA)

T. ATTIVITA', DISGIUNTE

- INSERT OR UPDATE ATTIVITA'

- ERROR = EXIST (SELECT * FROM ATTIVITA'

WHERE UTENTE = NEW.UTENTE AND ID <> NEW.ID

AND (INIZIOATT, FINEATT) OVERLAPS (NEW.INIZIOATT, NEW.FINEATT))

Risposta alla Domanda 7 (segue)

T. ATTIVITA'. INIZIO DOPPO NASCITA

- INSERT OR UPDATE ATTIVITA'

- ERROR = EXIST (SELECT * FROM UTENTE
WHERE USERNAME = NEW_UTENTE AND DATA_NASC > CAST(NEW_INIZIOATT AS DATE))

T. POSIZIONE. ORDINE

- INSERT OR UPDATE POSIZIONE

- OK = EXIST (SELECT * FROM POSIZIONE
WHERE PERC = NEW_PERC AND ORD = NEW_ORD - 1)
OR (NEW_ORD = 0)

T. POSIZIONE. ORDINE ORA

- INSERT OR UPDATE POSIZIONE

- ERROR = EXIST (SELECT * FROM POSIZIONE
WHERE PERC = NEW_PERC AND
(ORD > NEW_ORD AND ISTPOS < NEW_ISTPOS) OR (ORD < NEW_ORD AND ISTPOS > NEW_ISTPOS))

T. POSIZIONE. IN ATTIVITA'

- INSERT OR UPDATE POSIZIONE

- OK = EXIST (SELECT * FROM PERCORSO p, ATTIVITA' a
WHERE p.ID = NEW_PERC AND p.ATT = a.ID
AND a.INIZIOATT <= NEW_ISTPOS AND a.FINEATT >= NEW_ISTPOS)

T. SFIDA. SINGOLO

- INSERT OR UPDATE SFIDA

- OK = EXIST (SELECT * FROM SFIDA
WHERE ID = NEW_SFIDA AND TIPO = 'SING')

T. SFIDA. SQUADRA

- INSERT OR UPDATE SQUADRA

- OK = EXIST (SELECT * FROM SFIDA
WHERE ID = NEW_SFIDA AND TIPO = 'SQUA')

T. AMICO. NO DOPPIA

- INSERT OR UPDATE AMICO

- ERROR = EXIST (SELECT * FROM AMICO
WHERE UTENTE1 = NEW_UTENTE1 AND UTENTE2 = NEW_UTENTE2)

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

MAXSPORTAMICO($u_1: \text{STRINGA}, u_2: \text{STRINGA}, i: \text{TIMESTAMP}, f: \text{TIMESTAMP}$): $\text{STRINGA}[0..*]$

OK = EXIST (SELECT * FROM AMICO
WHERE (UTENTE1 = u_1 AND UTENTE2 = u_2) OR (UTENTE1 = u_2 AND UTENTE2 = u_1))

$R = \text{WITH } Q \text{ AS} \left(\text{SELECT SPORT, SUM(FINEATT - INIZIOATT)} \text{ TEMPO} \text{ FROM ATTIVITA'} \text{ WHERE INIZIOATT} \geq i \text{ AND FINEATT} \leq f \text{ GROUP BY SPORT} \right)$

SELECT SPORT FROM Q
WHERE TEMPO = (SELECT MAX(TEMPO) FROM Q)

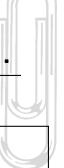
RETURN R

RESOCONTOSPORT($u: \text{STRINGA}, i: \text{TIMESTAMP}, f: \text{TIMESTAMP}$): $(\text{STRINGA, INTERVAL, REALGE})[0..*]$

OK = $i \leq f$ AND EXIST (SELECT * FROM ATTIVITA
WHERE UTENTE = u)

$R = \text{SELECT a.SPORT, SUM(a.FINEATT - a.INIZIOATT)} \text{ TEMPO, SUM(p.DIST())} \text{ DISTANZA} \text{ FROM ATTIVITA a LEFT OUTER JOIN PERCORSO p ON a.ID = p.ATT} \text{ AND a.INIZIOATT} \geq i \text{ AND a.FINEATT} \leq f \text{ GROUP BY a.SPORT}$

RETURN R



Risposta alla Domanda 8 (segue)

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.]

OPERAZIONI:

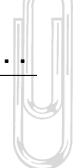
PERCORSO.DIST(): REALEGEZ

ERROR: NOT EXIST (SELECT * FROM POSIZIONE
WHERE PERC = this.ID)

R := WITH Q AS (SELECT p1.ORD, p2.ORD, DISTANZA(p1.LAT, p2.LAT, p1.LONG, p2.LONG) DIST
FROM POSIZIONE p1, POSIZIONE p2
WHERE p1.ORD = p2.ORD - 1)

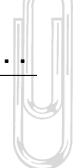
SELECT SUM(DIST) FROM Q.

RETURN R



[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.]