

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).

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

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

ALCENA Manno

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:



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

SPORTIVO/CREANTE

- NAME: String [e.g.]
- NAME: String
- Cognome: String
- GENERE: GENERE
- DATA N: DATE
- Peso kg: Int >= [e.g. 1]
- ALTEZZA CM: Int >= [e.g. 1]

ATTIVITA'

- INIZIO: DATA/DATE
- FINO: DATA/DATE
- DIFFICOLTÀ: PIFF [e.g. 1]

SFIDA

- NAME: STRING
- VALIDA DA: DATE
- VALIDA A: DATE
- MINIMO Min: Int >= 0
- CAT: CATEGORIA

SPORT

- NAME: String [e.g.]

PERSONA

DISCI

POVERO

- CAT: REAL >=
- LONG: REAL >=
- IST: DATA/DATE [e.g.]
- ORDINE: Int > 1 [e.g. 1]

SPADA SQUADRE

- MAX VP: Int >= 0

SQUADRA

- NAME: String

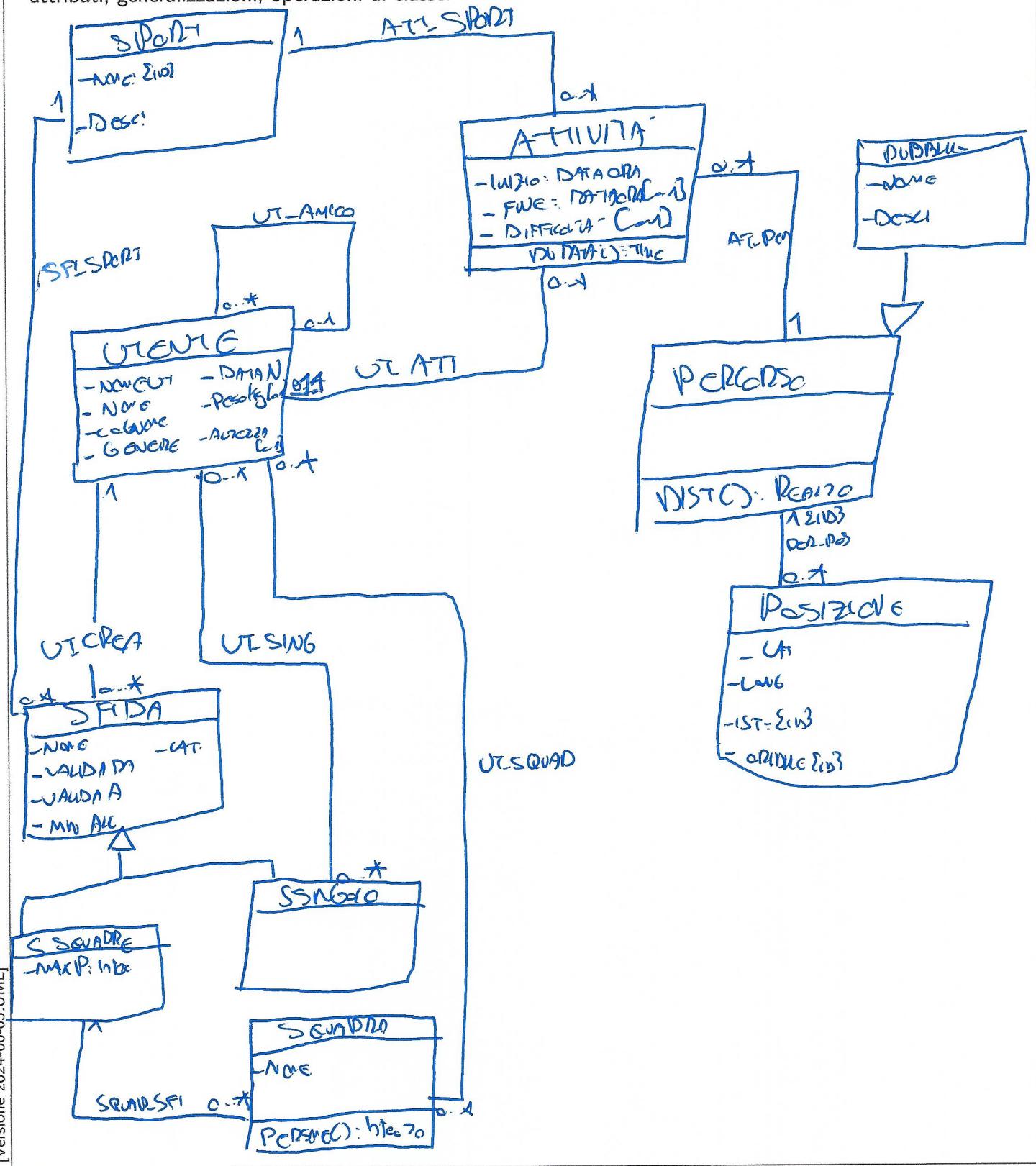
Risposta alla Domanda 1 (segue)

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

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



Risposta alla Domanda 2 (segue)

Matricola:

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

- Definire la specifica formale di eventuali operazioni necessarie a modellare i requisiti contrassegnati dalla barra laterale, ed eventuali vincoli esterni. Usare la logica del primo ordine estesa con teoria degli insiemi e semantica di mondo reale vista nel corso, usando il seguente alfabeto:
 - Un simbolo di predicato $C/1$ per ogni classe C .
Semantica di $C(x)$: x è una istanza di C .
 - Un simbolo di predicato $T/1$ per ogni tipo di dato T .
Semantica di $T(x)$: x è un valore di T .
 - Un simbolo di predicato $\text{assoc}/2$ per ogni associazione binaria assoc.
Semantica di $\text{assoc}(c_1, c_2)$: (c_1, c_2) è una istanza di assoc.
 - Un simbolo di predicato $\text{attr}/2$ per ogni attributo attr di entità
Semantica di $\text{attr}(c, v)$: uno dei valori dell'attributo attr dell'istanza c è v .
 - Un simbolo di predicato $\text{attr}/3$ per ogni attributo attr di associazione binaria.
Semantica di $\text{attr}(c_1, c_2, v)$: uno dei valori dell'attr. attr del link (c_1, c_2) è v .
 - Un simbolo di predicato $\text{op}/(n+2)$ per ogni operazione di classe ad n argomenti.
Semantica di $\text{op}(c, \arg_1, \dots, \arg_n, v)$: 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 $\text{adesso}/0$, interpretato come il valore del dominio DataOra che rappresenta l'istante corrente.

Risposta

<p>1 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ATTIUTA</p> <p>Operazioni, vincoli:</p> <p>① $[V. \text{ATTIUTA}. \text{CONTINUA}]$</p> <p>$\forall A, F$ $[\text{ATTIUTA}(A) \wedge \text{INZIO}(A, I) \wedge \text{FINE}(A, F)] \rightarrow \text{LF}$</p> <p>② $[V. \text{ATTIUTA}. \text{PERIODI} \text{ (cont)}]$</p> <p>$\forall A, I, F, P, \text{Pos}, \text{Is}$ $[\text{ATTIUTA}(A) \wedge \text{INZIO}(A, I) \wedge \text{FINE}(A, F) \wedge \text{PERIODI}(P)]$ $\wedge \text{ATTIUTA PER}(A, P) \wedge \text{Pos, ZIOTC}(P) \wedge \text{PERIODO}(P)$</p>	<p>2 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ATTIUTA</p> <p>Operazioni, vincoli:</p> <p>$1 \text{ IS } (\text{Pos}, \text{Is}) \rightarrow \text{IS } \subseteq \text{Pos}$</p> <p>② $[V. \text{ATTIUTA}. \text{NASCITA}]$</p> <p>$\forall A, I, V, N$ $[\text{ATTIUTA}(A) \wedge \text{INZIO}(I, I) \wedge \text{VERGINE}(C) \wedge$ $\text{VERATT}(C, A) \wedge \text{DAMMI}(C, V)] \rightarrow \text{NLI}$</p>
--	---

<p>3 Tipo: Classe Associazione (cerchiare) SFIDA</p> <p>Nome:</p> <p>Operazioni, vincoli:</p> <p>④ [U. SFIDA.CANTINTA] $\forall s, v_d, v_a$ $[SFIDA(s) \wedge VALIDADA(s, v_d) \wedge VALIDADA(s, v_a)]$ $\rightarrow v_d \neq v_a$ [U. SFIDA.MAXP] T $\forall s, m, sq, p$ $[SFIDA(s) \wedge SQUADRA(sq) \wedge MAXID(sq, m) \wedge$ $SQUADRA(sq) \wedge SQUAD_SFI(sq, s)] \wedge$</p>	<p>6 Tipo: Classe Associazione (cerchiare) PESONE</p> <p>Nome:</p> <p>Operazioni, vincoli:</p> <p>→ $PESONE \in \{Squadra\} \rightarrow P \leq n$</p>
<p>4 Tipo: Classe Associazione (cerchiare) SQUADRA</p> <p>Nome:</p> <p>Operazioni, vincoli:</p> <p>⑤ [U. SQUADRA. UTENTE DI U] $\forall s, u, s_2, sf$ $[SQUADRA(s) \wedge UTENTE(u) \wedge UT_SQUAD(s, s_2) \wedge$ $SQUADRA(s_2) \wedge s_1 \neq s_2 \wedge SQUAD_SF(s, s_2) \wedge$ $SQUAD_SF(s_1, sf) \wedge SQUAD_SF(s_2, sf)] \rightarrow$ $UT_SQUAD(u, s_2)$</p>	<p>7 Tipo: Classe Associazione (cerchiare) SFIDA</p> <p>Nome:</p> <p>Operazioni, vincoli:</p> <p>⑥ [U. SFIDA.PDN] $\forall s, v, v_d, v_a$ $[SFIDA(s) \wedge UTENTE(v) \wedge VALIDADA(s, v_d) \wedge$ $VALIDADA(s, v_a) \wedge UT_SQUAD(v_d, v_a)] \rightarrow NLVUD$ $\wedge DISTANZA(v_d, v_a) \wedge UT_SQUAD(v_d, v_a)] \rightarrow NLVUD$ [U. SFIDA.PRIMN] $\forall s, u, v_a, n$ $[SINGolo(s) \wedge VALIDADA(s, v_a) \wedge UTENTE(u) \wedge$</p>
<p>5 Tipo: Classe Associazione (cerchiare) DAMO</p> <p>Nome:</p> <p>Operazioni, vincoli:</p> <p>⑦ [U. SFIDA.DAMN] $\forall s, v_d, u, sq, n$ $[SQUADRA(s) \wedge VALIDADA(s, v_d) \wedge UTENTE(u) \wedge$ $DAMO(u, n) \wedge SQUADRA(sq) \wedge UT_SQUAD(u, sq) \wedge$ $SQUAD_SF(sq, s)] \rightarrow NLVUD$</p>	<p>8 Tipo: Classe Associazione (cerchiare) UTENTE</p> <p>Nome:</p> <p>Operazioni, vincoli:</p> <p>⑧ [U. UTENTE.NC AUTOAMICIZIA] $\forall u$ $(UTENTE(u) \rightarrow \exists v \text{ UT_AMICO}(u, v))$</p>

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

UTI = EPL DI VINCOLI

⑩ [U. ATIMITA.NC COMPE] T

$\forall A_1, U, A_2, l^1, l^2, F_1, F_2$

$[ATIMITA(A_1) \wedge GENERE(U) \wedge ATIMITA(A_2) \wedge INIZIO(A_1, l^1) \wedge INIZIO(A_2, l^2) \wedge FW \in (l^1, p) \wedge FINO(l^2, p)]$
 $\wedge ULATI(U, A_1) \wedge ULATI(U, A_2) \wedge A_1 \neq A_2 \rightarrow (l^1 \geq l^2 \vee l^2 \geq l^1)$

⑪ [U. POSIZIONE. SUCCESSIONE]

$\forall P, S_1, S_2, o_1, o_2$

$[POSIZIONE(P) \wedge ISZ(P, S_1) \wedge ISZ(P, S_2) \wedge ORDINE(P, o_1) \wedge ORDINE(P, o_2)] \rightarrow$

$\rightarrow [(S_1 < S_2 \wedge o_1 < o_2) \vee (S_2 < S_1 \wedge o_2 < o_1)]$

⑫ [U. POSIZIONE. SUCCESSIONE]

$\forall P, S_1, S_2, o_1, o_2$

$[POSIZIONE(P) \wedge ISZ(P, S_1) \wedge ISZ(P, S_2) \wedge ORDINE(P, o_1) \wedge ORDINE(P, o_2)] \rightarrow [(o_1 < o_2 \wedge S_1 < S_2) \vee (o_2 < o_1 \wedge S_2 < S_1)]$

TIPI DI DATO

- DIFF = {0..5}
- CAT = {"piu' km", "piu' aumenti"} enum
- GENERE: {"maschio", "femmina"} enum

OPERAZIONI DI CLASSE

CLASSE SQUADRA

- PERSONE () : Intero

• DUE: Nessuna

• DAT: M_{in} = M_{out}

SIA $P = \{U | ULATI(U) \wedge UT-SCUOLA(U, THE) \wedge SSQUADRA(U) \}$

RESULT = IP

Risposta alla Domanda 2 (segue)

- CLASSE ATIVITA'
- DUMTA (IN: DATAOM, FN: DATAFN): TIME
- o DUE = IN ≤ FN
- o POST = Min = Max
- Result = FN - IN

CLASSE PEDOC=PSO

→ DIST(L) : REAL > 0

o PED = NESSUNA

o Post: Min = Max

Sia $P = \{P_1, P_2, x_1, x_2, y_1, y_2\}$

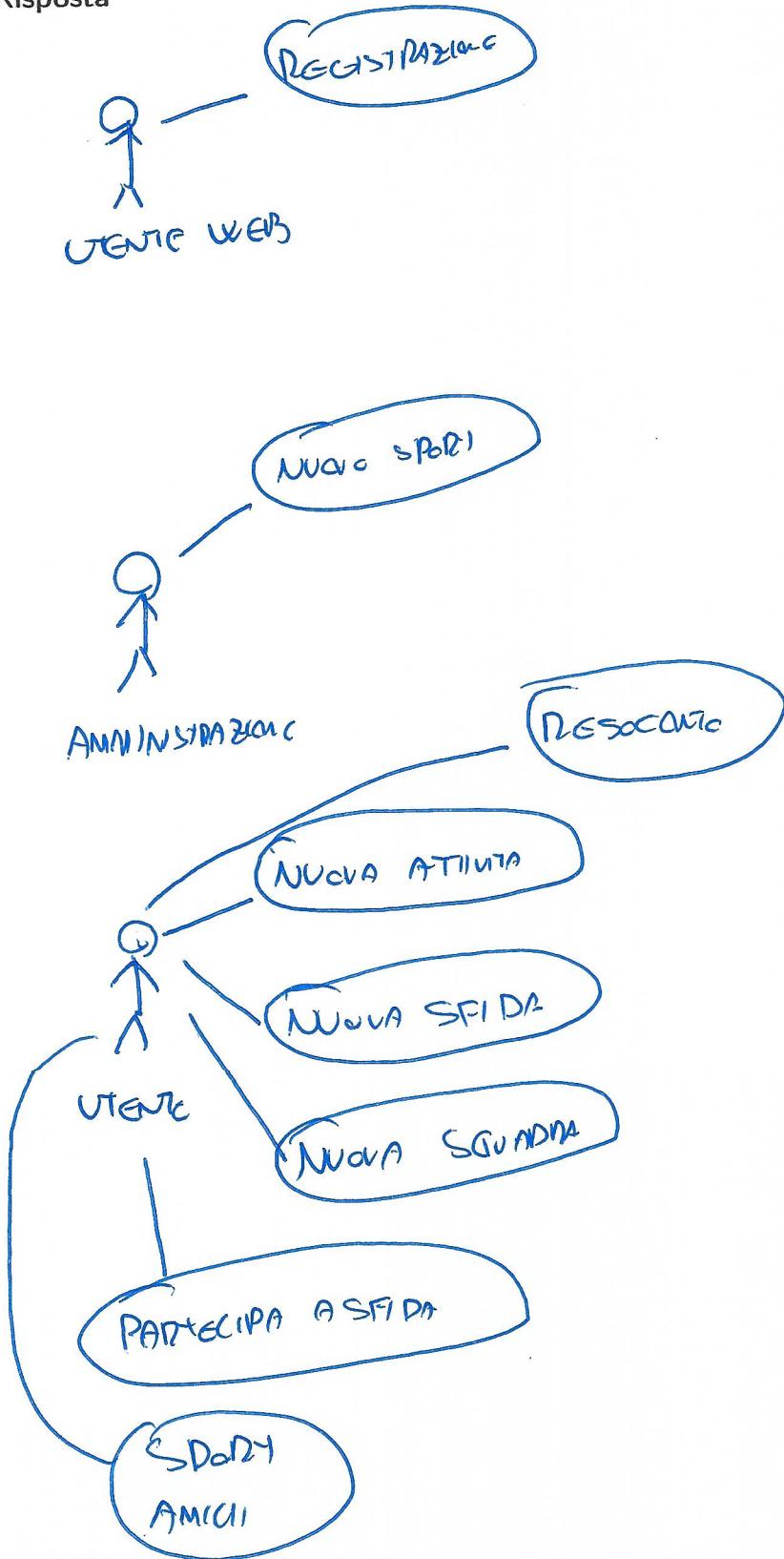
$\left. \begin{array}{l} \text{Posizione}(P_1) \wedge \text{Posizione}(P_2) \wedge \text{LAT}(P_1, x_1) \wedge \text{LAT}(P_2, x_2) \\ \wedge \text{LONG}(P_1, y_1) \wedge \text{LONG}(P_2, y_2) \wedge [Q_1, Q_2 \text{ ODO}(P_1, Q_1) \\ \wedge \text{ODO}(P_2, Q_2) \wedge Q_1 = Q_2] \wedge \text{PED_PO}(th_s, P_1) \wedge \text{PED_PO}(th_s, P_2) \end{array} \right\}$

$$TOT = \sum_{(P_1, P_2, x_1, x_2, y_1, y_2)} \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

$$Result = TOT$$

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 la segnatura delle operazioni in ogni use-case.

Risposta

USE CASE REGISTRAZIONE
 - REGISTRAZIONE (NU: String, N: String, C: String, G: Generale, DATA: DATE, DESO: Int [0..1], AC: Int [0..1]): Utente

USE CASE NUOVA SPORTESSA
 - INIZIERSPORT (N: String, D: String): Sport

USE CASE NUOVA ATTIVITA'
 - NUOVA_ATTIVITA' (D: DIFF [0..1]): ATTIVITA'
 - TERMINA_ATTIVITA' (A: ATTIVITA'): ATTIVITA'

USE CASE NUOVA SFIDA
 NUOVA_SFIDA (N: String, UD: DATA, VA: DATA, MUAC: Int [0..2], CAT: Cat, SQ: Bool): SFIDA

USE CASE NUOVA SQUADRA
 NUOVA_SQUADRA (U: String, S: SSquadre): SQUADRA

USE CASE PARTECIPA A SFIDA
 PARTECIPA (SS: SSquadra [-1]; SG: SQUADRE [-1], SQUADRA [-1])

USE CASE RESOCARO
 RESOCARO (U: DATA, FN: DATA): [SPORT, TIME, REALE] (Cat)

USE CASE SPORT AMICI
 SPORT_AMICI (IN: DATA, FIN: DATA): SPORT (Cat)

$$STIMATO = 95 \text{ Ma}$$

$$EFF = 66 \text{ Ma}$$

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 assume lo stesso vocabolario definito alla Domanda 2.

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

Risposta

Risposta SB01 ($S, S_{p=0}, T, IN, DATC, FW, DIRC$) $\rightarrow [T, T_{IN}, DATC, FW]$ [o-a]

SIA UN'AZIONE CHE INVIA IL USE CASE

$PDE = IN \in FUN$

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

su R: $\{ (S, T, D) \mid \begin{array}{l} ATTRA(S) \wedge REPORT_ATR(S, A) \wedge UL_ATR(U, A) \wedge INTR(U, I) \wedge FUF(A, F) \\ \wedge IN \in FUN, F \in DUDATA(A, T) \wedge PENDING(P) \wedge AL_PER(A, P) \end{array} \}$

¶

$$TOTTEMPO = \sum_{CT, DE, TR} T$$

$$TOTDIR = \sum_{G, M, P} D$$

$$R_{out} = (TOTTEMPO, TOTDIR)$$

Risposta (IN, DATC, FW, DIRC) = [S, PDI, T, R_{out}]

Pre: $IN \in FW$

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

$R = \{ (S, T, D) \mid \begin{array}{l} S \in T \wedge ATTRA(S) \wedge ULATT(U, T) \wedge ART-SB01(A, S) \\ \wedge \text{Desaccia} \rightarrow PDI(S, IN, FW, T, D) \end{array} \}$

$$R_{out} = R$$

2 Progettazione della base dati e delle funzionalità

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

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

Descrivere brevemente le principali scelte effettuate.

DBMS da utilizzare *PostgreSQL*.....

Corrispondenza tra tipi di dato concettuali e domini supportati dal DBMS

CREATE DOMAIN String AS VARCHAR (check value NOT NULL)

CREATE DOMAIN IntegeR AS Integer (check value NOT NULL AND value > 0)

CREATE DOMAIN IntegeR_0 AS Integer (check value NOT NULL AND value >= 0)

CREATE DOMAIN Real AS Real (check value NOT NULL AND value >= 0)

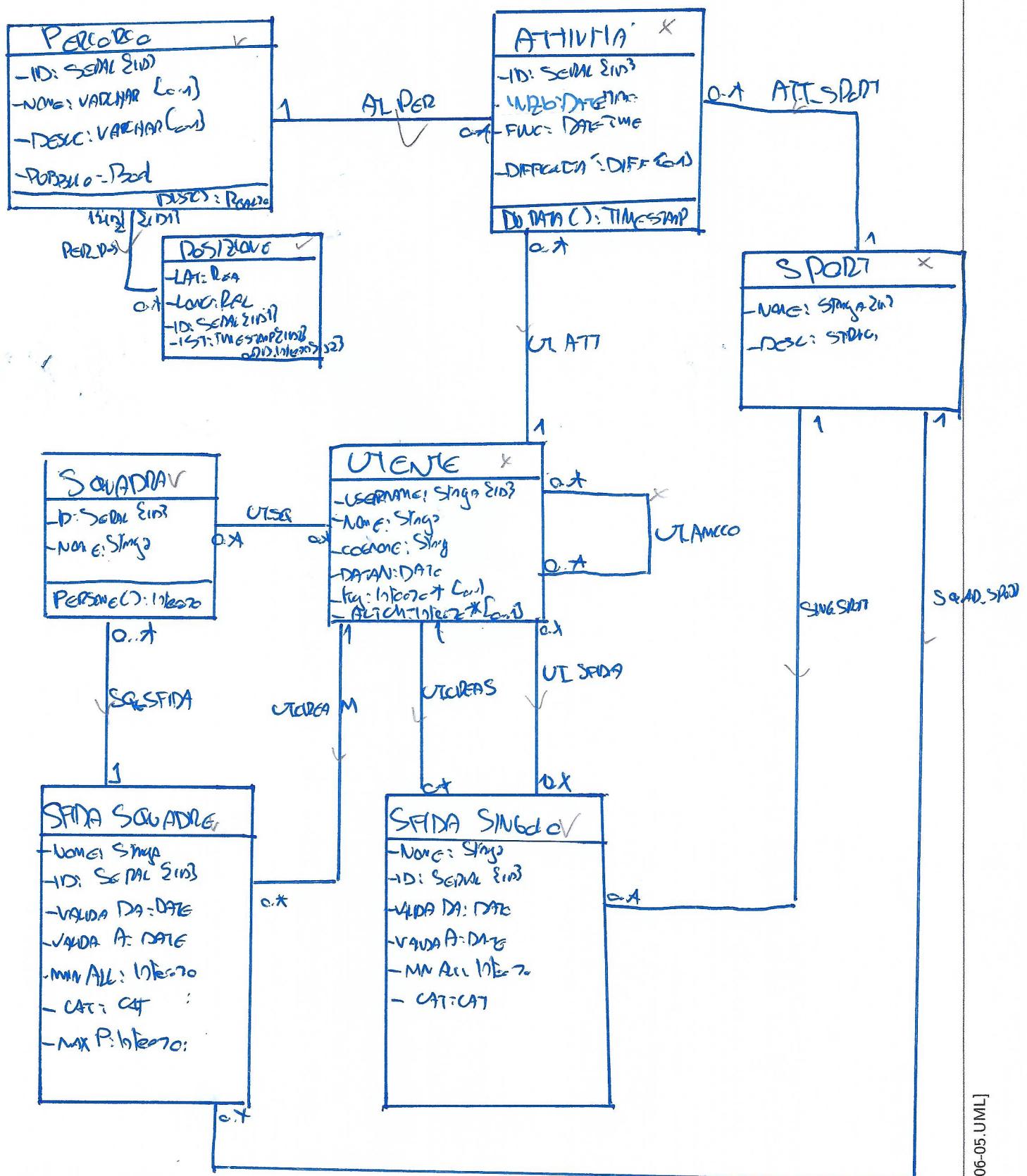
CREATE DOMAIN Boolean AS Boolean (check value IN (0, 1))

CREATE DOMAIN DIFF AS Integer (check value <= 100)

CREATE TYPE SEZIONE AS ENUM = { 'più Kilotruci', 'più AVENAMENTI' }

CREATE TYPE GENERE AS ENUM = { 'Maschi', 'Femmine' }

Diagramma UML delle classi ristrutturato



Breve descrizione delle scelte effettuate durante la ristrutturazione

- Ha preso PERCORSO PUBBLICO IN PERCORSO
- DIVISO LA GENERALIZZAZIONE DELLE SPES
- AGGIUNTO UN IDENTIFICATORE DANE MANCANTE

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. PERCORSO, PUBBLICO, TRUE] T

$\forall P, P_{OB}, N_D$

$[P \in \text{PERCORSO}(P), \text{PUBBLICO}(P, P_{OB}) \wedge \text{PUBBLICO}(P_{OB}, 'true') \wedge \text{NAME}(P, N) \wedge \text{DESC}(P, D)] \rightarrow (D = Nuc \wedge N = Nuc)$

[V. PERCORSO, PUBBLICO, FALSE]

$\forall P, P_{OB}, N_D$

$[P \in \text{PERCORSO}(P) \wedge \text{PERCORSO}(P, P_{OB}) \wedge \text{PUBBLICO}(P_{OB}, 'false') \wedge \text{NAME}(P, N) \wedge \text{DESC}(P, D)] \rightarrow (D = Nuc \wedge N = Nuc)$

Risposta alla Domanda 6 (segue)

Sono da modificare i vincoli di connivenza sulla sfida
 $\exists v, s, sq, v_d, v_a$
 $\exists s, sq \text{ SFIDA} \sqsubseteq \text{ADDS}(sq) \wedge \text{VALIDA}(sq, v_d) \wedge \text{VALIDA}(sq, v_a) \rightarrow v_d = v_a$

$\exists v, s, sq, v_d, v_a$
 $\exists s, sq \text{ SFIDA} \sqsubseteq \text{SING}(sq) \wedge \text{VALIDA}(sq, v_d) \wedge \text{VALIDA}(sq, v_a) \rightarrow v_d = v_a$

$\forall v, n, sq, v_d$
 $\forall v, n, sq \text{ SFIDA} \sqsubseteq \text{SCVADRE}(sq) \wedge \text{C1-CREAM}(v, sq) \wedge \text{VALIDA}(sq, v_d) \rightarrow v_d = v_a$

Stessa cosa per sfida da calcio.

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

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

1 Relazione	<u>SPOZI</u> (nome)	Derivante da:	<u>classe</u>	<u>associazione</u> (cerchiare)
Attributi	<u>Nome</u>				
Domini	<u>String</u>				

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

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

La relazione accorda le relazioni che implementano le seguenti associazioni:

2 Relazione	<u>UTENTE</u> (nome)	Derivante da:	<u>classe</u>	<u>associazione</u> (cerchiare)
Attributi	<u>ID</u>	<u>Nome</u>	<u>Cognome</u>	<u>DATAN</u>	<u>Possibilità</u> <u>Altro..?</u>
Domini	<u>Serial</u>	<u>String</u>	<u>String</u>	<u>Date</u>	<u>Intero</u> <u>Intero</u>

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

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

La relazione accorda le relazioni che implementano le seguenti associazioni:

3 Relazione	<u>UTENTI_AVCI</u> (nome)	Derivante da:	<u>classe</u>	<u>associazione</u> (cerchiare)
Attributi	<u>UTENTE1</u>	<u>UTENTE2</u>			
Domini	<u>Intero</u>	<u>Intero</u>			

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

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

FK: UTENTE1 REFERENCES UTENTE(1) check UTENTE1=UTENTE2

FK: UTENTE2 REFERENCES UTENTE(1)

La relazione accorda le relazioni che implementano le seguenti associazioni:

4 Relazione	<u>ATTIVITA</u> (nome)	Derivante da:	<u>classe</u>	<u>associazione</u> (cerchiare)
Attributi	<u>ID</u>	<u>Nome</u>	<u>DIFICOLTÀ</u> <u>Ora</u>	<u>SINT</u>	<u>PerfOps</u>
Domini	<u>Serial</u>	<u>DateTime</u>	<u>DateTime</u>	<u>Intero</u>	<u>String</u> <u>Intero</u>

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

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

FK: UTENTE REFERENCES UTENTE(1) FK: PEDOPS REFERENCES PERIOD(1)

FK: SPORT REFERENCES SPOR(1)

La relazione accorda le relazioni che implementano le seguenti associazioni: UTATT, ATT_SPORT, ATT_Period

5 Relazione	<u>PEDOPS</u> (nome)	Derivante da:	<u>classe</u>	<u>associazione</u> (cerchiare)
Attributi	<u>ID</u>	<u>Nome*</u>	<u>Cognome*</u>	<u>DURATA</u>	
Domini	<u>Serial</u>	<u>String*</u>	<u>String*</u>	<u>Bool</u>	

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

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

La relazione accorda le relazioni che implementano le seguenti associazioni:

6 Relazione <u>PERSON</u> (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>ID</u> <u>NAME</u> <u>LONG</u> <u>IST</u> <u>ADR</u> <u>PERDOS</u>	
Domini <u>SEMA</u> <u>PERDOS</u> <u>PERGEO</u> <u>DATI</u> <u>LIBRO</u> <u>BIBL</u>	
Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *	
Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):	
FK: <u>PERDOS REFERENCES PERDOS (11)</u> <u>UNIQUE (IST, PERDOS)</u> <u>UNIQG (ADR, PERDOS)</u>	
La relazione accorda le relazioni che implementano le seguenti associazioni: <u>PER-ADS</u>	

7 Relazione <u>SPADA_SALDO</u>... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>ID</u> <u>NAME</u> <u>VARIATA</u> <u>VARIATA</u> <u>MVAL</u> <u>CAT</u> <u>SPDR</u> <u>CRONC</u>	
Domini <u>SEMA</u> <u>STADIA</u> <u>DATI</u> <u>DATI</u> <u>LIBRO</u> <u>CAT</u> <u>STADIA</u> <u>LIBR</u>	
Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *	
Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):	
FK: <u>CRONC REFERENCES CRONC (11)</u> <u>SPDR REFERENCES STADIA (11)</u>	
La relazione accorda le relazioni che implementano le seguenti associazioni: <u>CRONC S... USNG_SHP</u>	

8 Relazione <u>UT_SPADA</u>.... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>LIQUID</u> <u>SPDR</u>	
Domini <u>INTESA</u> <u>LIBRO</u>	
Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *	
Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):	
FK: <u>UT_SPADA REFERENCES UT_UT (11)</u> <u>SPDR REFERENCES SPDR (11)</u>	
La relazione accorda le relazioni che implementano le seguenti associazioni:	

9 Relazione <u>SPADA_SQUADRA</u> (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>ID</u> <u>NAME</u> <u>VARIATA</u> <u>VARIATA</u> <u>MVALE</u> <u>CAT</u> <u>MAD</u> <u>CRONC</u> <u>SPDR</u>	
Domini <u>SEMA</u> <u>STADIA</u> <u>DATI</u> <u>DATI</u> <u>LIBRO</u> <u>CAT</u> <u>LIBRO</u> <u>INTESA</u> <u>LIBR</u>	
Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *	
Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):	
FK: <u>CRONC REFERENCES CRONC (11)</u> <u>SPDR REFERENCES SPDR (11)</u>	
La relazione accorda le relazioni che implementano le seguenti associazioni: <u>UT_OGRAM.. ISWAD SPDR</u>	

10 Relazione <u>SQUADRA</u> (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>ID</u> <u>NAME</u> <u>SEDA</u>	
Domini <u>SEMA</u> <u>STADIA</u> <u>LIBRO</u>	
Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *	
Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):	
FK: <u>SPDR REFERENCES SPDR_SQUADRA (11)</u>	
La relazione accorda le relazioni che implementano le seguenti associazioni: <u>SQUADRA</u>	

11 Relazione <u>UT_Squad</u> (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>Livello</u> <u>Squadra</u>	
Domini <u>Integ</u> <u>Intez</u>	
Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *	
Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio): FK: UT_GIUGNE REFERENCIOS UT_TECNICI FK: SQUADRA REFERENCIAS SAVANA(1)	
La relazione accorda 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 accorda 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 accorda le relazioni che implementano le seguenti associazioni:	

14 Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi	
Domini	
Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *	
Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):	
La relazione accorda le relazioni che implementano le seguenti associazioni:	

15 Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi	
Domini	
Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *	
Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):	
La relazione accorda le relazioni che implementano le seguenti associazioni:	

Ulteriori vincoli esterni

Per ogni ulteriore vincolo esterno (non ancora espresso perché non definibile mediante vincoli di chiave, foreign key, ennupla, dominio, inclusione), progettare un trigger che lo implementi, definendo: (a) gli eventi da intercettare (inserimento, modifica, eliminazione di ennupla); (b) quando intercettare tali eventi (appena prima o subito dopo l'evento intercettato); (c) la relativa funzione in pseudo-codice con SQL immerso che implementa il controllo del vincolo.

T. PERCORSO_PUBBLICO TRUE

INSEGNAMENTO o MODIFICA IN PERCORSO

O PER OPERAZIONE
 ISCEDID := (EXISTS (SELECT D.ID
 FROM PERCORSO P
 WHERE P.ID=NEWDID AND P.PUBBLICO = TRUE AND (P.NON=NUL OR P.DESCRNUL)))

IF ISGRADO:
 REGISTRA GRADO
 REVERSI

ELSE:
 COMMIT

T. SFIDA SQUADRE. CONTINUA'

INSEGNAMENTO o MODIFICA IN SFIDA SQUADRA

O PER OPERAZIONE

ISVALID := (NOT EXISTS (SELECT SQ.ID
 FROM SFIDA_SQUADRE SQ
 WHERE SQ.VALIDATA > SQ.VALIDATA AND SQ.ID=NEWID))

IF IS VALID:
 COMMIT

ELSE!
 GENERA ERRORE
 REVERSI

Risposta alla Domanda 7 (segue)

T. SFIDA SQUADRE MAX P

o INSERIMENTO o MODIFICA IN SFIDA SQUADRE

BLOCCO AD ID ZIA

IS ERRORE := (EXISTS (SELECT SQ.ID
 FROM SFIDA_SQUADRE SQ, EQUIPPI S
 WHERE NEW.ID = SQ.ID AND S.SFIDA = SQ.ID
 AD < PERSONE() > SQ.MAX_P))

IF IS ERRORE: GENERA ERRORE

REVERT

ELSE: COMITI

T. ATTIVITA' NO CONTEMPORANEE

o INSERIMENTO o MODIFICA IN ATTIVITA'

o DUE OPERAZIONI:

IP ERRORE := (EXISTS (SELECT A1.ID
 FROM ATTIVITA A1, ATTIVITA A2, UTENTE U
 WHERE A1.ID = NEW.ID AND A1.ID > A2.ID
 AND A1.UTENTE = U.ID AND A2.UTENTE = U.ID
 AND ((A1.INIZIO < A2.INIZIO & A1.FINE > A2.FINE) OR
 (A2.INIZIO < A1.INIZIO & A2.FINE > A1.FINE)))

IF IS ERRORE: GENERA ERRORE

REVERT

ELSE: COMITI

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 SQUADRA

- o PERSONE (S: Integer, SF: Integer): Intero

CREATE FUNZIO.

Q =
$$\left\{ \begin{array}{l} \text{SELECT COUNT(DISTINCT U.ID)} \\ \text{FROM SQUADRA S, SFIDASQUADRE SF, UTENTE U, UTENSILIA US} \\ \text{WHERE S.ID = S AND SF.ID = SF AND S.SFID = SF.ID} \\ \text{AND US.SQUADRA = S.ID AND US.UTENTE = U.ID} \end{array} \right\}$$

Result = Q

- CLASSE ATTIVITA'

- o DURATA (A.Integer): TIME STAMP

CREATE FUNZIO.

Q =
$$\left\{ \begin{array}{l} \text{SELECT (A.FINE - A. INIZIO)} \\ \text{FROM ATTIVITA A} \\ \text{WHERE A.ID = A} \end{array} \right\}$$

Result = A

CLASSE PRODUTO

- o DSY (P: Integer): Data

CREATE FUNZIO.

Risposta alla Domanda 8 (segue)

$$Q = \left(\begin{array}{l} \text{SELECT SUM(SQRT(SUM((POS1.LAT - POS2.LAT)^2 + (POS1.LONG - POS2.LONG)^2)))} \\ \text{FROM PERC_PSC P, POSIZIONE P1, POSIZIONE P2} \\ \text{WHERE P.ID = P1 AND P.ID = P2 AND P1.PERSONA = P2.PERSONA} \\ \text{AND POS1.ORDINE = POS2.ORDINE - 1} \\ \text{AND POS1.ORDINE = POS2.ORDINE - 1} \end{array} \right)$$

Result = Q

USE CAD

- Ricerca Collego (IN: DATE, FW: DATE):

MG.METRICA IN & UT (QUERI)

$$\text{SELECT AT.SPORTI AS SPORTE, SUM(SELECT AT.DURATA() AS TOTALE, SUM(SELECT D.DIST) FROM ATTIVITA AT WHERE AT.SPORTI = SPORTE) AS DISTANZA}$$

FROM ATTIVITA AT, PERSONA P
 WHERE AT.PERSONA = P.ID AND AT.VIENTE = 0 AND
 IN AT.INZIC AND FIN AT.FINE

Result = Q

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

-SPCRE AWCI (C. libri, in Date, FW: Date)

MCMODDE IN Q LA ALENDI

TABTEMP = (SELECT AT. SPRT AS SPRT, SUM (SELECT ATTIVITATI)) AS SPRT
 FROM ATTIVITA AT
 WHERE AT. SPRT = SPRT

FROM ATTIVITA AT, UTENTE U, MENSE AWCI UA
 WHERE U.ID = AND UA.UTENTE = U.ID AND UA.OTTOBRE = AT.UTENTE
 AND AT. Mese IN (M. FEB, MAR)
 GROUP BY SPRT

Q = (SELECT SPRT
 FROM TABTEMP
 WHERE OTTOBRE = (SELECT Mese (cnd)
 FROM TABTEMP))

Risposta