

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

Dati dello studente e dell'esame

Cognome e nome: BIANCO SIMONE Matricola: //

Data: 19/08/23

Corso di laurea e canale di appartenenza:

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

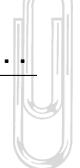
Firma di un membro della Commissione per
avvenuta identificazione:
.....

Rinuncia alla prova

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



Questo modulo è ottimizzato per la stampa fronte-retro



Istruzioni e regole d'esame

Prima dell'esame

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

Durante l'esame

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

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

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

Al momento della consegna

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

In caso di rinuncia

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

Sommario delle domande

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

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

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

- Analisi dei dati (45 minuti; 75 minuti al massimo):
 - un diagramma ER concettuale (*)
 - il relativo dizionario dei dati
 - le specifiche dei domini concettuali non di tipo base
 - eventuali vincoli esterni, espressi utilizzando il linguaggio della logica del primo ordine (*)
- Analisi delle funzionalità:
 - un diagramma UML degli use-case (5 minuti; 10 minuti al massimo)
 - la segnatura di tutte le operazioni di use-case (10 minuti)
 - la specifica delle operazioni di use-case necessarie a modellare i requisiti contrassegnati dalla barra laterale (come quella qui a sinistra) in termini di precondizioni e postcondizioni, utilizzando il linguaggio della logica del primo ordine (*) (30 minuti; 60 minuti al massimo)

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

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

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

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

Le pagine da 33 in poi possono essere utilizzate per scrivere minute che non verranno valutate.



Questa pagina è stata intenzionalmente lasciata vuota

1 Analisi concettuale

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

Risposta

SPORTIVO

- NICKNAME
- NOME, COGNOME
- SFSSO
- NASCITA
- PESO (0,1)
- ALTEZZA (0,1)
- POSSONO ESSERE AMICI

SPORT

- NOME
- DESC

ATTIVITA'

- UTENTE
- SPORT
- INIZIO
- FINE (0,1)
- DIFFICOLTÀ (0,1)

PERCORSO

- INSERIRE AI COORD
- PUÒ ESSERE PUBBLICO
- SE PUBBLICO HA UN NOME E UN COMM

COORD

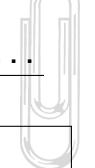
- LAT
- LONG

SFIDA

- SQ / SING
- PIÙ KM / PIÙ ALT
- NOME
- INIZIO, FINE
- DURATA - ALT
- SPORTE (0,N)
- SE A SQ HA MAX - PART. SQ

SVIADRA

- NOME
- SPAZIO
- COMPONENTI



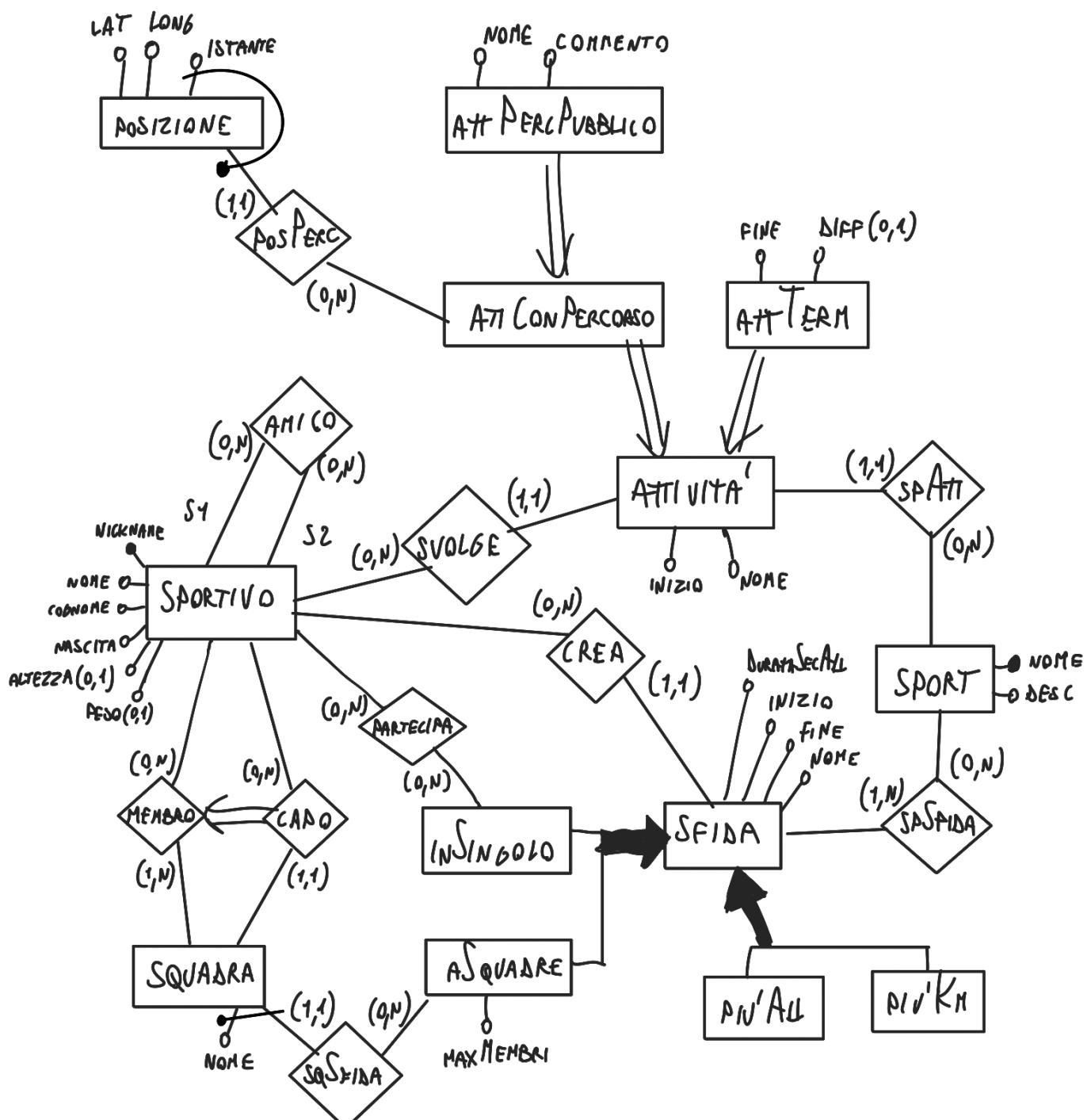
Risposta alla Domanda 1 (segue)

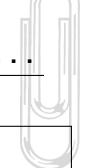
Domanda 2 (45 minuti; 75 minuti al massimo) Proseguire la fase di Analisi Concettuale dei requisiti, producendo un diagramma ER concettuale per l'applicazione, il dizionario dei dati ed eventuali vincoli esterni.

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: POSIZIONE</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>attributo</th><th>dominio</th><th>moltep. (*)</th></tr> </thead> <tbody> <tr> <td>LAT</td><td>REAL 7,0</td><td></td></tr> <tr> <td>LONG</td><td>REAL 7,0</td><td></td></tr> <tr> <td>ISTANTE</td><td>DATAORA</td><td></td></tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p>	attributo	dominio	moltep. (*)	LAT	REAL 7,0		LONG	REAL 7,0		ISTANTE	DATAORA		<p>[2] Tipo: Entità Relationship (cerchiare)</p> <p>Nome: ATT PERC PUBBLICO</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>attributo</th><th>dominio</th><th>moltep. (*)</th></tr> </thead> <tbody> <tr> <td>COMMENTO</td><td>STR</td><td></td></tr> <tr> <td>NAME</td><td>STR</td><td></td></tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p>	attributo	dominio	moltep. (*)	COMMENTO	STR		NAME	STR	
attributo	dominio	moltep. (*)																				
LAT	REAL 7,0																					
LONG	REAL 7,0																					
ISTANTE	DATAORA																					
attributo	dominio	moltep. (*)																				
COMMENTO	STR																					
NAME	STR																					

<input type="checkbox"/> 3	Tipo: Entità	Relationship (cerchiare)
Nome:	SPORTIVO	
attributo	dominio	moltep. (*)
NICKNAME	STR	
NOME	STR	
COGNOME	STR	
SESSO	{M, F}	
NASCITA	DATAORA	
PESO	REALIZZO	(0,1)
ALTEZZA	REALIZZO	(0,1)

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

Vincoli:

- V. SPORTIVO. CONTINUITÀ ATTIVITÀ
 $\forall t, a, n, i \quad \text{SPORTIVO}(t) \wedge \text{SVOLGE}(t, a) \wedge$
 $\text{NASCITA}(a, n) \wedge \text{INIZIO}(a, i) \rightarrow n < i$
- V. SPORTIVO. CONTINUITÀ SFIDA
 $\forall t, a, a', n, i \quad \text{SPORTIVO}(t) \wedge \text{SFIDA}(t, a') \wedge$
 $\text{NASCITA}(a, n) \wedge \text{INIZIO}(a', i) \rightarrow$
 $(\text{PARTECIPA}(t, a) \vee (\exists a'' \text{MEMBRO}(a, a'') \wedge$
 $n \leq \text{SFIDA}(a', a'')) \rightarrow n < i$

<input type="checkbox"/> 5	Tipo: Entità	Relationship (cerchiare)
Nome:	ATTIVITÀ	
attributo	dominio	moltep. (*)
NAME	STR	
INIZIA	DATAORA	

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

Vincoli:

- V. ATTIVITÀ. CONTINUITÀ
 $\forall a, i \quad \text{ATTIVITÀ}(a) \wedge \text{INIZIA}(a, i) \rightarrow$
 $(\forall t \text{ FINE}(t, i) \rightarrow i < t)$
- V. ATTIVITÀ. NSCONTIN
 $\forall a, t, a', i, i' \quad \text{ATTIVITÀ}(a) \wedge \text{SVOLGE}(t, a) \wedge$
 $a \neq a' \wedge \text{INIZIO}(a, i) \wedge \text{INIZIO}(a', i') \rightarrow$
 $\exists t' \text{ DATAORA}(t) \wedge (i \leq t \wedge (\forall t'' \text{ FINE}(t, t'') \rightarrow t < t''))$
 $\wedge (i' \leq t' \wedge (\forall t''' \text{ FINE}(t', t''') \rightarrow t' < t'))$

<input type="checkbox"/> 4	Tipo: Entità	Relationship (cerchiare)
Nome:	ATT TERM	
attributo	dominio	moltep. (*)
FINE	DATAORA	
DIFF	INTERO[1,5]	(0,1)

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

Vincoli:

<input type="checkbox"/> 6	Tipo: Entità	Relationship (cerchiare)
Nome:	SPORT	
attributo	dominio	moltep. (*)
NOME	STR	
DESC	STR	

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

Vincoli:

7	Tipo: Entità Relationship (cerchiare)
	Nome: ... SFIDA
attributo	dominio
NAME INIZIO FINE DURATA	STR DATAORA DATAORA INTEROZERO
(*) solo se diversa da (1,1)	

Vincoli:

V. SFIDA. CONTINUITÀ

$$\forall i, j, k \quad SFIDA(i) \wedge INIZIO(i, j) \wedge FINE(j, k) \rightarrow i < k$$

9	Tipo: Entità Relationship (cerchiare)
	Nome: ... A SQUADRE
attributo	dominio
MAX MEMBRI	INTEROZERO
(*) solo se diversa da (1,1)	

Vincoli:

V. A SQUADRE. MAXMEMBRI

$$\forall s, sq, m \quad A SQUADRE(s) \wedge SQ SFIDA(sq, s) \wedge MAX MEMBRI(sq, m) \rightarrow \left| \{ u \mid \begin{array}{l} UTENTE(u) \wedge \\ MEMBRO(u, sq) \end{array} \} \right| \leq m$$

8	Tipo: Entità Relationship (cerchiare)
	Nome: ... SQUADRA
attributo	dominio
NAME	STR
(*) solo se diversa da (1,1)	

10	Tipo: Entità Relationship (cerchiare)
	Nome: ... AMICO
attributo	dominio
(*) solo se diversa da (1,1)	

Vincoli:

V. SQUADRA. SOLO UNA SQUADRA

$$\forall u, s, s', s'' \quad UTENTE(u) \wedge MEMBRO(u, s) \wedge SFIDA(s) \wedge SQ SFIDA(s', s) \wedge SQ SFIDA(s'', s') \rightarrow \neg MEMBRO(u, s')$$

Vincoli:

V. AMICO. NO SUO AMICO

$$\forall u, u' \quad AMICO(u, u') \rightarrow u \neq u'$$

<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. (*)

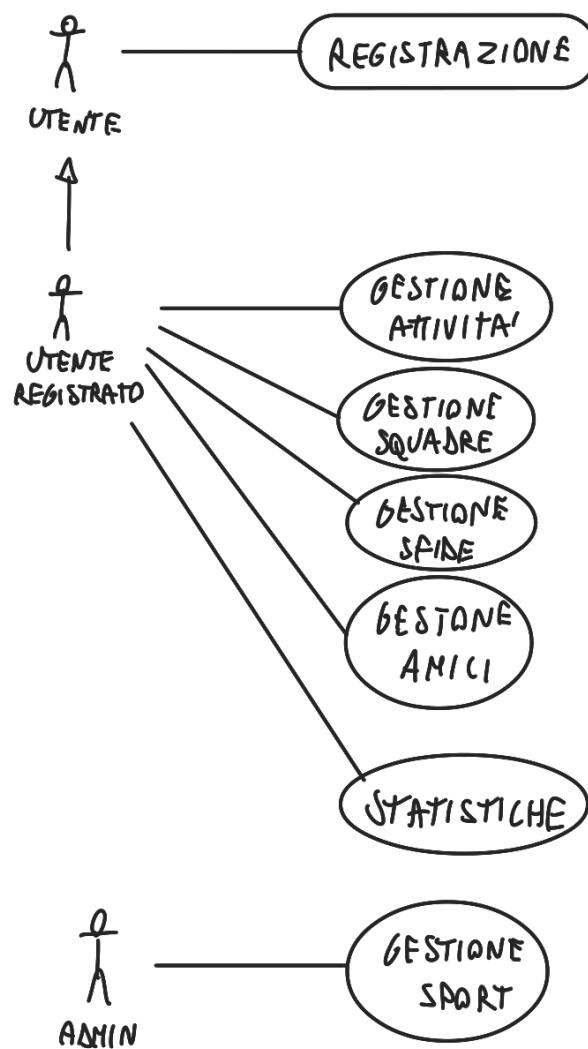
(*) solo se diversa da (1,1)

Vincoli:

Ulteriori vincoli esterni, specifica di eventuali operazioni ausiliarie invocate da tali vincoli, e specifica dei domini concettuali non di tipo base

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: REGISTRAZIONE (nome use-case)

Operazioni dello use-case:

REGISTRASPORTIVO ($n: \text{STRINGA}$, $n': \text{STRINGA}$, $c: \text{STRINGA}$, $d: \text{DATA}$,
 $z: \text{REALIZZO } (0,1)$, $p: \text{REALIZZO } (0,1)$): SPORTIVO

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

Operazioni dello use-case:

INSEGNIA ATTIVITA' ($a: \text{SPORTIVO}$, $s: \text{SPORT}$, $i: \text{DATAORA}$, $n: \text{STRINGA}$): ATTIVITA'

3 Specifica use-case: GESTIONE SQUADRE (nome use-case)

Operazioni dello use-case:

CREA SQUADRA ($u: \text{SPORTIVO}$, $n: \text{NONE}$, $sf: \text{SFIDA}$): SQUADRA

AGGIUNGI MEMBRO ($a: \text{SQUADRA}$, $u: \text{SPORTIVO}$)

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

Operazioni dello use-case:

CREA_SFIDA_IN_SINODO (u: SPORTIVO, n: STRINGA, s: SPORT(1,N), i: DATAORA, f: DATAORA) : INSINODO

CREA_SFIDA_ASQUARE (u: SPORTIVO, n: STRINGA, s: SPORT(1,N), i: DATAORA, f: DATAORA
m: INTERO>1) : INSINODO

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

Operazioni dello use-case:

AGGIUNGI_AMICO (u: SPORTIVO, u': SPORTIVO)

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

Operazioni dello use-case:

RESO_CONTATO_ALLENAMENTI (u: SPORTIVO, i: DATAORA, f: DATAORA) : (a: SPORT, t: INTERO>0, k: REALE>0)

SPORT_PIU_PRATICATO_AMICI (u: SPORTIVO, i: DATAORA, f: DATAORA) : SPORT(0,N)

CALCOLA_TEMPO_SPORT (u: SPORTIVO, a: SPORT) : INTERO>0

CALCOLA_KM_SPORT (u: SPORTIVO, a: SPORT) : REALE>0

CALCOLA_KM_ATIVITA' (a: ATTIVITA_PERCORSO) : REALE>0

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

Operazioni dello use-case:

INSERISCI_SPORT (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

CALCOLA KmAttività ($a : \text{ATT(ConPerCorso)}$): REALE ≥ 0

PRE:

POST:

$$\rho = \left\{ (p, p', d) \mid \begin{array}{l} \text{POSIZIONE}(c) \wedge \text{POSIZIONE}(c') \wedge \text{POSPERC}(z, c) \wedge \\ \wedge \text{POS PERC}(z, p) \wedge p \neq p' \wedge (\exists lt, lt', lg, lg', i, i' \\ \wedge \text{LAT}(p, lt) \wedge \text{LAT}(p', lt') \wedge \text{LONG}(p, lg) \wedge \text{LONG}(p', lg') \wedge \\ \wedge \text{DISTANZA}(lt, lg, lt', lg') = d \wedge \text{ISTANTE}(p, i) \wedge \text{ISTANTE}(p', i') \\ \wedge (\nexists p'', i'' \text{ POS PERC}(a, p'') \wedge \text{ISTANTE}(p', i'') \wedge i < i'' \wedge i'' < i') \end{array} \right\}$$

$$\text{RESULT} = \sum_{(p, p', d) \in \rho} d$$

CALCOLA KmSport ($u : \text{SMARTPHONE}$, $a : \text{SPORT}$, $i : \text{DATAORA}$, $x : \text{DATAORA}$): REALE ≥ 0

PRE: $i < x$

POST:

$$A = \left\{ (a, k) \mid \begin{array}{l} \text{ATT(ConPerCorso}(a) \wedge \text{CALCOLAKmAttività}(a) = k \wedge \text{SPORTATT}(a, a) \\ \wedge \text{SVOLGE}(u, a) \wedge (\exists i' \text{ INIZIO}(a, i') \wedge i \leq i' \wedge (\exists f' \text{ FINE}(a, f') \rightarrow i \leq f')) \end{array} \right\}$$

$$\text{RESULT} = \sum_{(a, k) \in A} k$$

CALCOLATEMAO Sport ($u : \text{SMARTPHONE}$, $a : \text{SPORT}$, $i : \text{DATAORA}$, $x : \text{DATAORA}$): INTERO ≥ 0

PRE: $i < x$

POST:

$$A = \left\{ (a, t) \mid \begin{array}{l} \text{ATTIVITA'}(a) \wedge \text{SVOLGE}(u, a) \wedge \text{SPORTATT}(a, a) \wedge \\ (\exists i \text{ INIZIO}(a, i) \wedge (\exists f' \text{ FINE}(a, f') \rightarrow t = f' - i \wedge i \leq f') \wedge \\ (\exists f'' \text{ FINE}(a, f'') \rightarrow t = \text{ADESSO} - i) \wedge i \leq i') \end{array} \right\}$$

$$\text{RESULT} = \sum_{(a, t) \in A} t$$

Risposta alla Domanda 5 (segue)

RESOCONTO ALLENAMENTI (u : SPORTIVO, i : DATAORA, f : DATAORA): $(a: \text{SPORT}, t: \text{INTERO}, o, k: \text{NATURALE} \geq 0)$

PRE: $i < f$

POST:

$$S = \left\{ (a, t, k) \mid \text{SPORT}(a) \wedge \text{CALCOLA}_k \text{SPORT}(u, i, f) = k \wedge \text{CALCOLA_TEMPO}_k \text{SPORT}(u, i, f) = t \right\}$$

RESULT = S

SPORT PIÙ PRATICATO AMICI (u : SPORTIVO, i : DATAORA, f : DATAORA): $\text{SPORT}(o, n)$

PRE: $i < f$

POST:

$$A = \left\{ a \mid \text{UTENTE}(a) \wedge (\text{AMICO}(u, a) \vee \text{AMICO}(a, u)) \right\}$$

$$S = \left\{ (a, t) \mid \text{SPORT}(a) \wedge t = \sum_{a \in A} \text{CALCOLA_TEMPO}_k \text{SPORT}(a, i, f) \right\}$$

$$S_{\max} = \underset{(a, t) \in S}{\text{ARGMAX}}(t)$$

$$\text{RESULT} = \left\{ a' \mid \exists t' \quad (a', t') \in S_{\max} \right\}$$

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.

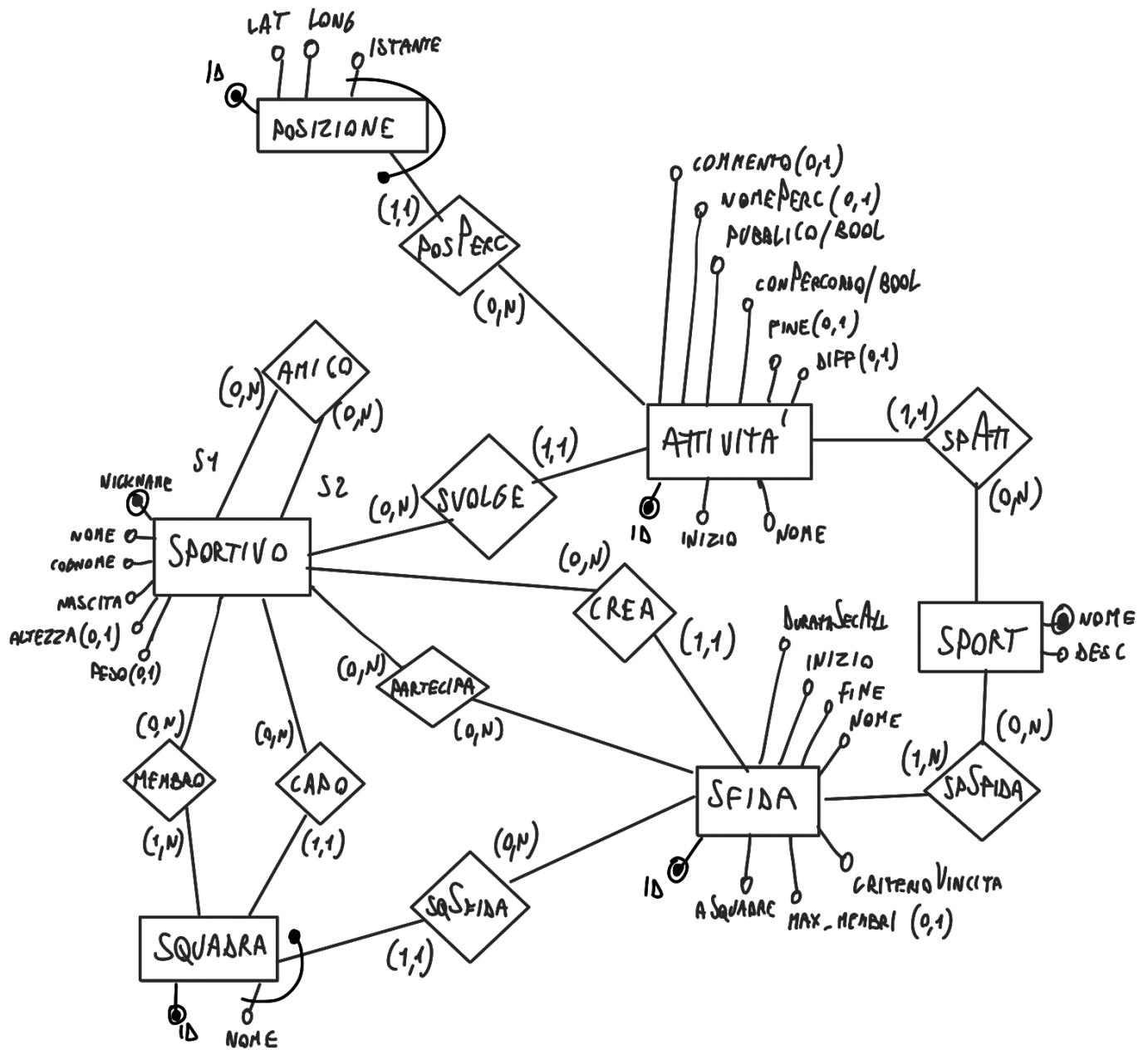
Descrivere brevemente le principali scelte effettuate.

DBMS da utilizzare **PostgreSQL**.....

Corrispondenza tra domini concettuali e domini supportati dal DBMS

- `CREATE DOMAIN INT-GZ AS INTEGER CHECK (VALUE >= 0)`
- `CREATE DOMAIN INT-GEZ AS INTEGER CHECK (VALUE >= 0)`
- `CREATE DOMAIN REAL-GEZ AS REAL CHECK (VALUE >= 0)`
- `CREATE DOMAIN INT-1-5 AS INTEGER CHECK (VALUE >= 1 AND VALUE <= 5)`
- `CREATE DOMAIN STR AS VARCHAR(200)`
- `CREATE TYPE SEX AS ENUM ('M','F')`
- `CREATE TYPE WIN-T AS ENUM ('PIU'ALL','PIU'KM')`

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)

X • V. ATTIVITA'. DIFF

$$\forall a \text{ ATTIVITA}'(a) \rightarrow (\exists d \text{ DIFF}(a, d) \rightarrow \exists f \text{ FINE}(d, f))$$

X • V. ATTIVITA'. NOME & COMMENTO

$$\forall a \text{ ATTIVITA}'(a) \rightarrow (\exists n, c \text{ NOME}(a, n) \wedge \text{COMMENTO}(a, c) \leftrightarrow \text{PUBBLICO}(a, \text{TRUE}))$$

X • V. ATTIVITA'. PUBBLICO

$$\forall a \text{ ATTIVITA}'(a) \rightarrow (\text{PUBBLICO}(a, \text{TRUE}) \rightarrow \text{CONPERCORSO}(a, \text{TRUE}))$$

T • V. ATTIVITA'. POSIZIONE

$$\forall a \text{ ATTIVITA}'(a) \rightarrow (\exists p \text{ POS_PERC}(a, p) \rightarrow \text{CONPERCORSO}(a, \text{TRUE}))$$

X • V. SFIDA. MAX-MARY

$$\forall s \text{ SFIDA}(s) \rightarrow (\exists m \text{ MAX_MEMBRI}(s, m) \leftrightarrow \text{ASQUADRE}(s, \text{TRUE}))$$

T • V. SFIDA. PARTECIPA

$$\forall s \text{ SFIDA}(s) \rightarrow (\exists u \text{ PARTECIPA}(s, u) \rightarrow \text{ASQUADRE}(s, \text{FALSE}))$$

Risposta alla Domanda 6 (segue)

T • V. SFIDA. SQFQUADRA

$$\forall s \quad SFIDA(s) \rightarrow (\exists a \quad SQFIDA(a, s) \rightarrow ASQUADRA(a, \text{TRUE}))$$

X • V. CAPO. ISA

$$\forall u, s \quad \text{SPORTIVO}(u) \wedge \text{CAPO}(u, s) \rightarrow \text{MEMBRO}(u, s)$$

T • V. ASQUADRA. MAXMEMBRI

$$\forall s, s', m \quad SPIDA(s) \wedge ASQUADRA(s, \text{TRUE}) \wedge SQSPIDA(s, s') \wedge \text{MAXMEMBRI}(s, m)$$

$$\rightarrow |\{u \mid \begin{array}{l} \text{UTENTE}(u) \wedge \\ \text{MEMBRO}(u, s') \end{array}\}| \leq m$$

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 ...<u>POSIZIONE</u>... (nome)	Derivante da: <u>entità</u> relationship (cerchiare)					
Attributi <u>ID</u> LAT LONG ISTANTE ATT						
Domini INTEGER REAL_GEZ REAL_GEZ TIMESTAMP 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):

SERIAL : ID **FK: (ATT) REF ATIVITA' (ID)**
CHIAVE: (ISTANTE, ATT)

La relazione accorda le relazioni che implementano le seguenti relationship: **POS/PERC**

2 Relazione ...<u>SPORTIVO</u>... (nome)	Derivante da: <u>entità</u> relationship (cerchiare)					
Attributi <u>NICKNAME</u> NOME COGNOME NASCITA ALTEZZA* PESO*						
Domini STR STR STR DATE REAL_BFZ REAL_BFZ						

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:

3 Relazione ...<u>AMICO</u>..... (nome)	Derivante da: <u>entità</u> relationship (cerchiare)					
Attributi <u>SPORTIVO1</u> <u>SPORTIVO2</u>						
Domini STR STR						

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: (SPORTIVO1) REF SPORTIVO(NICKNAME) **ENN: SPORTIVO1 ≠ SPORTIVO2**
FK: (SPORTIVO2) REF SPORTIVO(NICKNAME)

La relazione accorda le relazioni che implementano le seguenti relationship:

4 Relazione ...<u>SQUADRA</u>..... (nome)	Derivante da: <u>entità</u> relationship (cerchiare)					
Attributi <u>ID</u> NOME CAPO SFIDA						
Domini INTEGER STR STR 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: (CAPO) REF SPORTIVO(NICKNAME) **CHIAVE: (NOME, SFIDA)** **FK: (CAPO, ID) REF MEMBRO(SQ, SQ)**
FK: (SFIDA) REF SFIDA(ID) **INC: (ID) ⊆ MEMBRO(SQ)**

La relazione accorda le relazioni che implementano le seguenti relationship: **CAPO, SQ_SFIDA**

5 Relazione ...<u>MEMBRO</u>..... (nome)	Derivante da: <u>entità</u> relationship (cerchiare)					
Attributi <u>SQ</u> <u>SQ</u>						
Domini STR 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: (SQ) REF SFIDA(ID)
FK: (SQ) REF SQUADRA(ID)

La relazione accorda le relazioni che implementano le seguenti relationship:

6 Relazione ...SFIDA..... (nome)	Derivante da: entità relationship (cerchiare)					
Attributi <u>ID</u> NOME INIZIO FINE ASQUADRE MAXMEMBRI* CREATORE NICKNAME ALL						
Domini INTEGER STR TIMESTAMP TIMESTAMP BOOLEAN INTEGER STR INT-6EZ						
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):						
SERIAL: ID		DOM: MAXMEMBRI > 1		ENN: ASQUADRE = TRUE \Leftrightarrow MAXMEMBRI \neq NULL		
ENN: INIZIO < FINE		FK: (CREATORE) REF SPORTIVO(NICKNAME)		INC: (ID) \subseteq SFIDA(SFIDA)		
La relazione accorda le relazioni che implementano le seguenti relationship: ... (cerchiare)						

7 Relazione ...SPORT..... (nome)	Derivante da: entità relationship (cerchiare)					
Attributi <u>NOME</u> DESCRIZIONE						
Domini STR STR						
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: ... (cerchiare)						

8 Relazione ...SP_SFIDA..... (nome)	Derivante da: entità relationship (cerchiare)					
Attributi <u>SFIDA</u> <u>SPORT</u>						
Domini INTEGER STR						
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: ... (cerchiare)						

9a Relazione ...ATTIVITA'..... (nome)	Derivante da: entità relationship (cerchiare)										
Attributi <u>ID</u> NOME SPORTIVO SPORT INIZIO FINE* DIFF* CONPERCORSO											
Domini INTEGER STR STR STR TIMESTAMP TIMESTAMP INT-1..5 BOOLEAN											
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):											
SERIAL: ID		ENN: FINE \neq NULL \rightarrow INIZIO < FINE		ENN: DIFF \neq NULL \rightarrow FINE \neq NULL							
FK: (SPORTIVO) REF SPORTIVO(NICKNAME)											
FK: (SPORT) REF SPORT(NOME)											
La relazione accorda le relazioni che implementano le seguenti relationship: ... (cerchiare)											

9b Relazione ...ATTIVITA'..... (nome)	Derivante da: entità relationship (cerchiare)					
Attributi PUBBLICO* NOMEPERC* COMMENTO*						
Domini BOOLEAN STR STR						
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):						
ENN: PUBBLICO = TRUE \rightarrow CONPERCORSO = TRUE						
ENN: NOMEPERC \neq NULL \wedge COMMENTO \neq NULL \leftrightarrow PUBBLICO = TRUE						
La relazione accorda le relazioni che implementano le seguenti relationship: ... (cerchiare)						

11 Relazione .PARTECIPA....(nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>SPORTIVO</u> <u>SPIDIA</u>	
Domini STR 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: (SPORTIVO) REF SPORTIVO (NICKNAME)
FK: (SPIDIA) REF SPIDA (/Δ)

La relazione accorda le relazioni che implementano le seguenti relationship:

12 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:

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 ennuple); (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. ATTIVITA'. POSIZIONE

- INSERIMENTO IN POSIZIONE
- PRE - OPERAZIONE

$$\text{ISVALID} = (\text{EXISTS}(\text{SELECT } \text{x} \text{ FROM ATTIVITA' z} \\ \text{WHERE } z.\text{ID} = \text{NEW.ID} \text{ AND z.CONFERMATO} = \text{TRUE}))$$

```
IF ISVALID
    COMMIT
ELSE
    GENERA ERRORE
```

• T. SFIDA. SQUADRA

- INSERIMENTO IN SQUADRA
- PRE - OPERAZIONE

$$\text{ISVALID} = (\text{EXISTS}(\text{SELECT } \text{x} \text{ FROM SPIDER S} \\ \text{WHERE } s.\text{ID} = \text{NEW.SFIDA} \text{ AND s.ASQUADRE} = \text{TRUE}))$$

```
IF ISVALID
    COMMIT
ELSE
    GENERA ERRORE
```

• T. SFIDA. PARTECIPA

- INSERIMENTO IN PARTECIPA
- PRE - OPERAZIONE

$$\text{ISVALID} = (\text{EXISTS}(\text{SELECT } \text{x} \text{ FROM SPIDER S} \\ \text{WHERE } s.\text{ID} = \text{NEW.SFIDA} \text{ AND s.ASQUADRE} = \text{FALSE}))$$

```
IF ISVALID
    COMMIT
ELSE
    GENERA ERRORE
```

Risposta alla Domanda 7 (segue)

• T. ASQUAARE. MAXMEMBRI

- INSERIMENTO IN MEMBRO

- PRE-OPERAZIONE

$M = (\text{SELECT } S.\text{MAXMEMBRI AS MAX FROM SFIDA } S, \text{SQUADRA } SQ \text{ WHERE } S.ID = SQ.SFIDA \text{ AND } SQ.ID = NEW.SQUADRA)$

$C = (\text{SELECT COUNT(*) AS C FROM MEMBRO } M \text{ WHERE } M.SQUADRA = NEW.SQUADRA)$

IF $C.C < M.M$

COMMIT

ELSE

GENERA ERRORE

• T. SQUADRA. SOLO UNA SQUADRA

- INSERIMENTO IN MEMBRO

- PRE-OPERAZIONE

$\text{IS ERROR} = (\text{EXISTS } (\text{SELECT } * \text{ FROM MEMBRO } M, \text{SQUADRA } SQ, \text{SQUADRA } SQ' \text{ WHERE } M.SQUADRA = SQ.ID \text{ AND } M.SPORTIVO = NEW.SPORTIVO \text{ AND } NEW.SQUADRA = SQ'.ID \text{ AND } SQ.SFIDA = SQ'.SFIDA))$

IF ISERROR

GENERA ERRORE

ELSE

COMMIT

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

- DB. CALCOLA_KM_ATIVITA' ($a: \text{INTEGER}$): REAL-GEZ

$$Q = \left(\begin{array}{l} \text{SELECT SUM(EXT.DISTANZA(P.LAT, P.LONG, P'.LAT, P'.LONG)) AS SUM} \\ \text{FROM POSIZIONE } P, \text{POSIZIONE } P', \text{ATIVITA' } a' \\ \text{WHERE } a'.ID = :a \text{ AND } a'.CONPERCORSO = \text{TRUE} \text{ AND} \\ p.ACT = :a \text{ AND } p'.ACT = :a \text{ AND NOT EXISTS(} \\ \text{SELECT } * \\ \text{FROM POSIZIONE } P'' \\ \text{WHERE } P''.ACT = :a \text{ AND } P''.ISTANTE <= P''.ISTANTE} \\ \text{AND } P''.ISTANTE <= P'.ISTANTE \end{array} \right)$$

RETURN Q.SUM

- DB. CALCOLA_KM_SPORT ($u: \text{STR}, a: \text{STR}, i: \text{TIMESTAMP}, f: \text{TIMESTAMP}$): REAL-GEZ

IF $:f < :i$
GENERA ERRORE

$$Q = \left(\begin{array}{l} \text{SELECT SUM(DB.CALCOLA_KM_ATIVITA'}(a.ID)) AS SUM} \\ \text{FROM ATIVITA' } a \\ \text{WHERE } a.SPORTIVO = :u \text{ AND } a.SPORT = :a \text{ AND} \\ :i <= a.INIZIO \text{ AND } (FINE IS NULL OR a.INIZIO <= :f) \\ \text{AND } a.CONPERCORSO = \text{TRUE} \end{array} \right)$$

RETURN Q.SUM

- DB. CALCOLA_TEMPO_SPORT ($u: \text{STR}, a: \text{STR}, i: \text{TIMESTAMP}, f: \text{TIMESTAMP}$): INT-BEZ

IF $:f < :i$
GENERA ERRORE

$$Q = \left(\begin{array}{l} \text{SELECT SUM(CASE WHEN } a.FINE IS NOT NULL} \\ \text{THEN } a.FINE - a.INIZIO \\ \text{ELSE CURRENT_TIMESTAMP - a.INIZIO END)} \\ \text{FROM ATIVITA' } a \\ \text{WHERE } a.SPORTIVO = :u \text{ AND } a.SPORT = :a \text{ AND} \\ :i <= a.INIZIO \text{ AND } (FINE IS NULL OR a.INIZIO <= :f) \end{array} \right)$$

RETURN Q.SUM

Risposta alla Domanda 8 (segue)

- RESO CONTO ALLENAMENTI ($u: \text{STR}, i: \text{TIMESTAMP}, f: \text{TIMESTAMP}$): INSIEME ($\langle a: \text{STR}, t: \text{INT-6EZ}, k: \text{REAL-6EZ} \rangle$)
IF $:f \leq :i$
GENERA ERRORE

$Q = (\text{SELECT } s.\text{NAME AS } a$
 $\quad \text{DB.CALCOLATETEMPOSPORT}(:u, s.\text{NAME}, :i, :f),$
 $\quad \text{DB.CALCOLAKMSPORT}(:u, s.\text{NAME}, :i, :f))$

FROM SPORT S)

NOTA: è inefficiente
(versione efficiente
a MINUTA 33)

RETURN Q

- SPORT PIU' PRATICATO AMICI ($u: \text{STR}, i: \text{TIMESTAMP}, f: \text{TIMESTAMP}$): INSIEME ($\langle a: \text{STR} \rangle$)

IF $:f \leq :i$
GENERA ERRORE

TMAPTABLE ($a: \text{STR}, t: \text{INT-6EZ}$)
FK: (a) REF SPORT (NAME)

INSERT INTO TMAPTABLE (a, t)

(SELECT s.NAME AS a, SUM(DB.CALCOLATETEMPOSPORT(u.NICKNAME, s.NAME, :i, :f)) AS t
 FROM Sportivo u, Sport S, Amico a
 WHERE (a.SPORTIVO1 = :u AND a.SPORTIVO2 = s.NICKNAME)
 OR (a.SPORTIVO2 = :u AND a.SPORTIVO1 = s.NICKNAME)
 GROUP BY a)

$Q = (\text{SELECT } a \text{ FROM TMAPTABLE}$
 $\text{WHERE TT.t} = (\text{SELECT MAX(TT.t) FROM TMAPTABLE TT}))$

RETURN 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.]

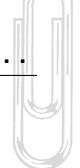
- **RESO CONTO ALLENAMENTI** ($u: \text{STR}, i: \text{TIMESTAMP}, f: \text{TIMESTAMP}$): $\text{INSIENE}(\langle a: \text{STR}, t: \text{INT_GE2}, k: \text{REAL_GE2} \rangle)$
*** VERSIONE EFFICIENTE**

If $:f < :i$
 GENERA ERRORE

$Q = (\text{SELECT } S.\text{NAME AS } a,$
 $\text{SUM}(\text{CASE WHEN } 2.\text{FINE IS NOT NULL}$
 $\text{THEN } 2.\text{FINE} - 2.\text{INIZIO}$
 $\text{ELSE CURRENT_TIMESTAMP} - 2.\text{INIZIO END}) \text{ AS } t,$
 $\text{SUM}(2.\text{CALCOLA_ATTIVITA'}(2.\text{IA})) \text{ AS } k$
 $\text{FROM ATTIVITA' } 2$
 $\text{WHERE } 2.\text{SPORTIVO} = :u \text{ AND } 2.\text{SPORT} = :t \text{ AND}$
 $:i \leq 2.\text{INIZIO} \text{ AND } (\text{FINE IS NULL OR } 2.\text{INIZIO} \leq :f))$

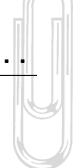
GROUP BY a)

RETURN $Q.\text{SUM}$



[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.]