ALLENAMONDO

SPORTIVO:

- NICKNAME : STR
- NOME : STR
- COGNOME : STR
- GENERE : {'M', 'F'}
- NASCITA : DATE
- PESO : FLOAT [0..1]
- ALTEZZA : FLOAT [0..1]
- AMICI

SPORT:

- NOME : STR
- DESCRIZIONE : STR

ALLENAMENTO:

- SPORT
- INIZIO : TIMESTAMP
- FINE : TIMESTAMP
- DIFFICOLTÀ : 1..5 [0..1]
- PERCORSO [0..1]

PERCORSO:

- TAPPE : [1..*]
- DESC : STR
- PUBBLICO : BOOL

TAPPE

- ISTANTE : TIMESTAMP
- LONG ~~COORDINATE~~ : FLOAT
- LAT

SFIDA:

- ~~TIPO : {'SINGOLO', 'SQUADRA'}~~ → SINGOLA:
  - PARTECIPANTI
- NOME : STR
- DURATA : ?
- CATEGORIA : {'+ KM', '+ ALLENAMENTI'} → A SQUADRA:
  - MAX : INT > 0
  - SQUADRE
- SPORT [1..*]
- GETVINCITORE()

SQUADRA:

- NOME
- COMPONENTI

~~KM:~~

~~- GETPUNTEGGIO()~~

~~ALL:~~

VINCOLI:

$\forall u \; UTENTE(u) \rightarrow \neg \exists S1,S2 \; SQUADRE\begin{matrix}(S1)\\(S2)\end{matrix} \wedge S1 \neq S2 \wedge PRT\begin{matrix}(S1,SF)\\(S2,SF)\end{matrix} \wedge APP\begin{matrix}(U,S1)\\(U,S2)\end{matrix}$

UNO SPORTIVO DEVE APPARTENERE AD UNA SINGOLA SQUADRA RISPETTO AD UNA CERTA SFIDA

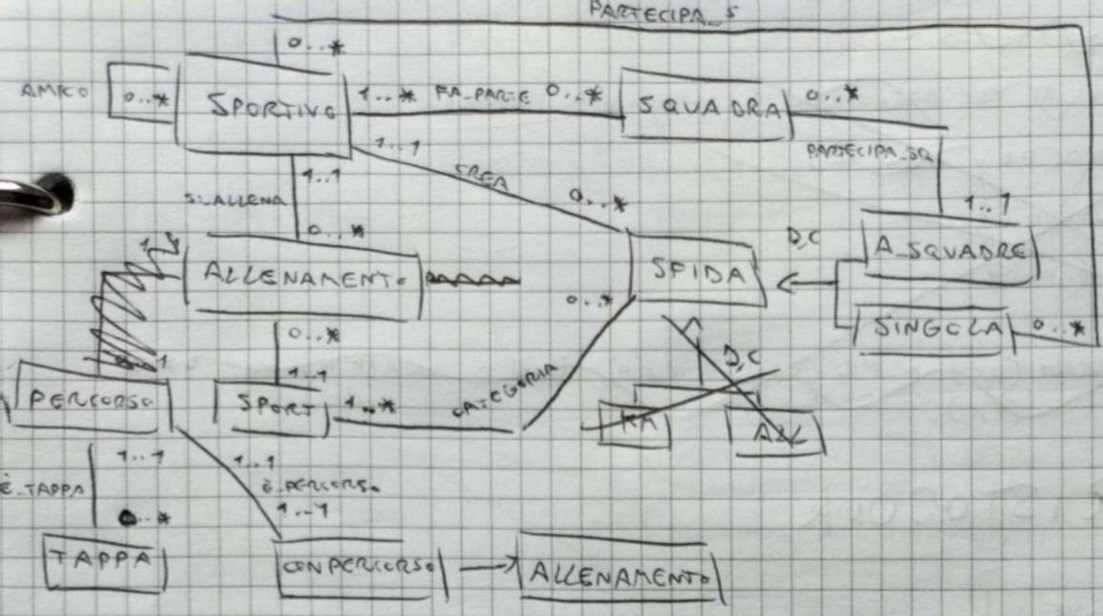
~~UT(U) ∧ USQ(U,S1) ∧ PRT(S1,SF) → ∃ S2 SQ(S2) ∧ USQ(U,S2) ∧ PRT(S2,SF)~~

~~SFIDA-SQ(SF) ∀ U, UTENTE(U) → ¬∃ S1,S2~~

FUNZ.:

- REGISTRAZIONE UTENTE
- INSERIMENTO SPORT
- 〃 ATTIVITÀ
- CREAZIONE SFIDA
- ISCRIZIONE A SFIDA

- TEMPO ALL.
- KM PERCORSI
- SPORT + PRATICO DA AMICI



VINCOLI:

- UN UTENTE NON PUÒ ~~NASCERE PRIMA~~ ALLENARSI PRIMA DI ESSERE NATO
- UGUALE PER LE SFIDE
- UN ALLENAMENTO NON PUÒ FINIRE PRIMA DI ESSERE INIZIATO, LO STESSO PER SFIDE
- LE TAPPE DI UN PERCORSO DEVONO RIENTRARE TEMPORALMENTE NELL'ALLENAMENTO
- UNA SQUADRA NON PUÒ AVERE PIÙ DI MAX COMPONENTI
- UN UTENTE PUÒ SVOLGERE UN SOLO ALLENAMENTO ALLA VOLTA
- LE TAPPE DI UN PERCORSO NON DEVONO SOVRAPPORSI TEMPORALMENTE

$\forall P, T1, T2, I1, I2\ PERCORSO(P) \wedge È\_TAPPA(P,T1) \wedge È\_TAPPA(P,T2) \wedge IST\,\begin{matrix}(T1,I1)\\(T2,I2)\end{matrix} \rightarrow I1 \neq I2$

- $\forall U, N, I, F\ \ (UTENTE(U) \wedge NASCITA(U,N) \wedge ALLENAMENTO(U,I) \rightarrow N < F$

  $\wedge\ ALLENAMENTO.\overset{}{\underset{INIZIO}{~~FINE~~}}(I,F))$

- $(\wedge PARTECIPA\_S(U,I) \wedge SINGOLA.INIZIO(I,F) \rightarrow N < F$

- $, S\ (\wedge FA\_PARTE(U,I) \wedge PARTECIPA\_SQ(I,F) \rightarrow N < S$

  $\wedge\ A\_SQUADRE.INIZIO(F,S))$

$\forall A, I, F\ \ ALLENAMENTO(A) \wedge INIZIO(A,I) \wedge FINE(A,F) \rightarrow I < F$

$SFIDA(A) \rightarrow I < F$

$\forall T, P, A, I, F, IS\ \ TAPPA(T) \wedge È\_TAPPA(T,P) \wedge È\_PERCORSO(P,A) \rightarrow I < IS \wedge IS < F$

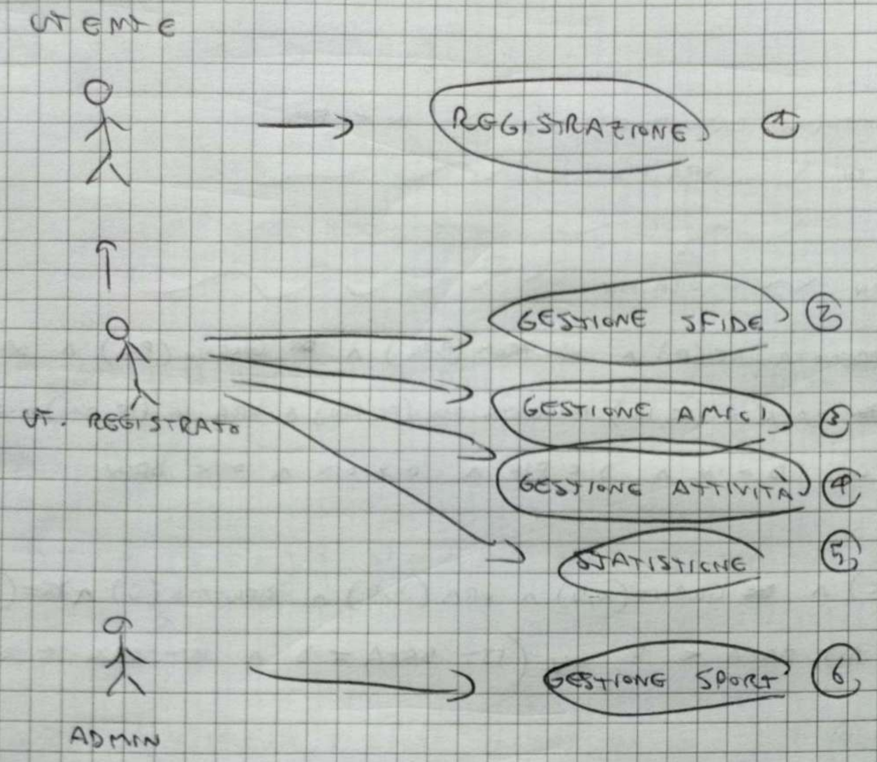
$\wedge\ INIZIO(A,I) \wedge FINE(A,F) \wedge ISTANTE(T,IS)$

$\forall S, M, N, P\ \ SQUADRA(S) \wedge \overset{PARTECIPA\_SQ}{~~MAX~~}(S,P) \wedge MAX(P,M) \rightarrow |N| \leq M$

$\wedge\ N = \{ U \mid UTENTE(U) \wedge FA\_PARTE(U,S) \}$

$\forall U, A1, A2, I1, I2, F1, F2\ \ UTENTE(U) \wedge SI\_ALLENA(U,A1) \wedge SI\_ALLENA(U,A2) \rightarrow F1 < I2 \vee F2 < I1$

$\wedge\ INIZIO(A1,I1) \wedge A1 \neq A2$



SPECIFICHE

(1) REGISTRAZIONE (NICK : STR, NOME : STR, COG : STR, GEN : SEX, NAS : DATE, PESO : REAL > 0 [0..1], AZT : ~~~ [~~]) : SPORTIVO

(2) CREA_SFIDA (NOME : STR, INIZ : DATE, FINE : DATE, CATE : CAT, M : INT > 0 [0..1]) : SFIDA

PART_SFIDA (S : SPORTIVO, SF : SFIDA_S)

ISCRIVITI_SQ (S : SPORTIVO, SQ : SQUADRA)

PART_SFIDA_SQ (S : SQUADRA, SF : SFIDA_SQ)

CREA_SQ (N : STR, SF : SFIDASQ) : SQUADRA

(3)

AGGIUNGI_AMICO (S1 : SPORTIVO, S2 : SPORTIVO)

GETTONATO (S : SPORTIVO, I : DATE, F : DATE) : SPORT [0, N]

(4) INSERISCI (S : SPORTIVO, SP : SPORT, I : DATE, F : DATE, D : INT [0..5], P : PERCORSO [0..1]) : ATTIVITÀ

(5) TEMPO_AZ (S : SPORTIVO) : INT

KM_PERCORSI (S : SPORTIVO) : INT

~~TEMPO_AZZ (S: SPORTIVO)~~

⑥ INS_SPORT (N : STR , D : STR) : SPORT

RESOCONTO_ALL , SPORT + PRAT , KMPERCORSI , ALL PATTI
① ② ③ ④

① RES_AZZ (U : SPORTIVO , IN : DATE , FIN : DATE) : $(\text{SPORT}, \overset{\text{REAL}}{\cancel{\text{INT}}} \geq 0, \text{REAL} \geq 0)$ $[0..*]$

PRE : $IN < FIN$

POST :

$S = \{ (S, T, K) \mid SPORT(S) \wedge TMP\_SPORT(U, S, IN, FIN) = T \wedge KM\_SPORT(U, S, IN, FIN) = K \}$

RETURN S

TMP_SPORT (U : SPORTIVO , S : SPORT , IN : DATE , FIN : DATE) : (REALE ≥ 0)

PRE : $IN < FIN$,

POST :

$A = \{ (A, T) \mid ATTIVITÀ(A) \wedge PRATICA(U, A) \wedge SP\_ATT(A, S) \wedge INIZIO(A, I) \wedge FINE(A, F) \wedge$
$IN \leq I \wedge FIN \geq F \wedge F - I = T \}$

RETURN $\sum_{(A,T) \in A} T$

KM_SPORT (U ~ , S ~ , IN ~ , FIN ~) : (REALE ≥ 0)

PRE : $IN < FIN$

POST :

$A = \{ (A, K) \mid CONPERCORSO(A) \wedge SP\_ATT(A, S) \wedge PRATICA(U, A) \wedge INIZIO(A, I) \wedge FINE(A, F) \wedge$
$IN \leq I$ $\wedge$ $FIN \geq F$ AND $KMATT(A) = K \}$

RETURN $\sum_{(A,K) \in A} K$

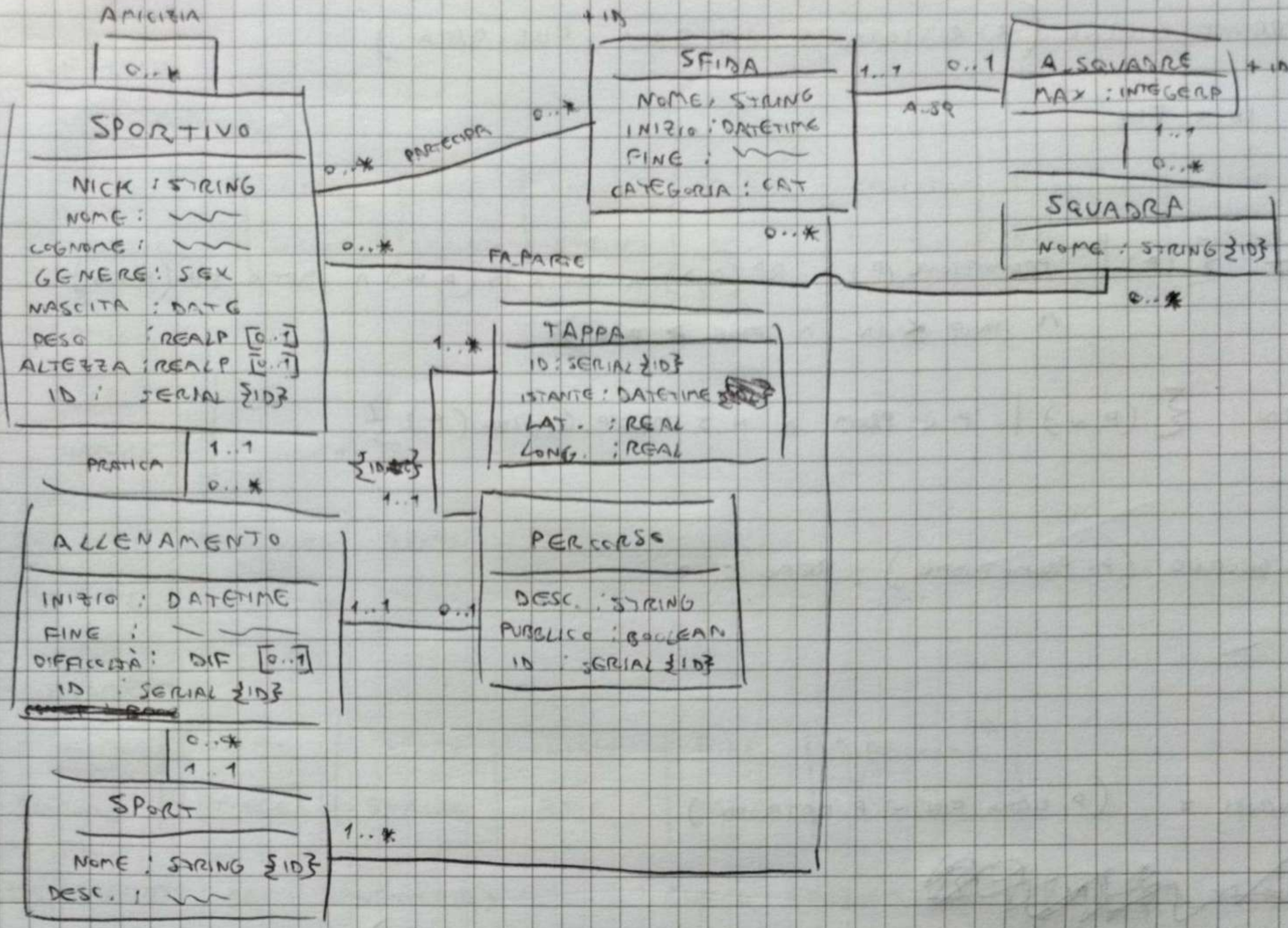
KMATT (A : CONPERCORSO) : (REALE ≥ 0)

PRE :

POST :

$POSIZ = \{ \overset{\#}{(P, P', LAT, LAT', LONG, LONG')} \mid TAPPA\overset{(P')}{(P)} \wedge LATITUDINE \overset{(P', LAT')}{(P, LAT)} \wedge LONGITUDINE \overset{(P', LONG')}{(P, LONG)} \wedge IST\overset{(P', I')}{(P, I)}$
$\wedge I \neq I' \wedge (\neg \exists P'', I'' \, (TAPPA(P'') \wedge IST(P'', I'') \wedge I < I'' \wedge I'' < I') \}$

RETURN $\sum_{\# \in POSIZ}$ ~~[illegible]~~ ~



$\forall$ S, SQ SFIDA(S) $\land$ A_SQ (S, SQ) $\rightarrow$ $\neg \exists$ SP SPORTIVO(SP) $\land$ PARTECIPA (SP, S)

CALC_TMP_AMICI (U : SPORTIVO , IN : DATE, FIN : DATE) : SPORT [0..*]

PRE : IN, FIN

POST :

AMICI = $\{$ (AM) $|$ SPORTIVO(AM) $\land$ (AMICO (U, AM) $\lor$ AMICO (AM, U)) $\}$

TMP = $\{$ (S, T) $|$ SPORT(S) $\land$ T = $\sum_{AM \in AMICI}$ TMP_SPORT (S, AM, IN, FIN) $\}$

~~MAX =~~ $\underset{(S,T) \in TMP}{\text{ARGMAX}}$ (T)

~~RETURN~~

SPORTIVO

~~SERIAL~~

ID	NICK	NOME	COGNOME	GENERE	NASCITA	PESO*	ALTEZZA*
SERIAL	STRING	STRING	STRING	SEX	DATE	REAL_P	REAL_P

AMICO

U1	U2
INTEGER	INTEGER

FK : U1 REFERENCES SPORTIVO (ID)

FK : U2 〃 〃

ALLENAMENTO

ID	UTENTE	SPORT	INIZIO	FINE	DIFFICOLTÀ*
SERIAL	INTEGER	STRING	DATE	DATE	DIF

FK UTENTE REF SPORTIVO (ID)

〃 SPORT 〃 SPORT (NOME)

CHECK (INIZIO < FINE)

PERCORSO

ID	DESCRIZIONE	PUBBLICO	ALLENAMENTO
SERIAL	STRING	BOOL	INTEGER

FK ALLENAMENTO REF ALLENAMENTO (ID)

UNIQUE (ALLENAMENTO)

TAPPA

ID	ISTANTE	LONG	LAT	PERCORSO
SERIAL	DATETIME	REAL	REAL	ID

FK PERCORSO -> PERCORSO (ID)

UNIQUE (ID, PERCORSO)

SFIDA

ID	NOME	INIZIO	FINE	CATEGORIA
SERIAL	FINE	DATE	DATE	CAT

CHECK (INIZIO ≤ FINE)

PARTECIPA

UTENTE	SFIDA
INTEGER	INTEGER

FK : UTENTE REF UTENTE (ID)

SFIDA REF SFIDA (ID)

SPORT_SFIDA

SFIDA	SPORT
INTEGER	STRING

FK SFIDA -> SFIDA (ID)

SPORT -> SPORT (NOME)

A_SQUADRE

SFIDA	MAX
INTEGER	INTEGER

FK SFIDA -> SFIDA (ID)

~~PARTECIPA_SQ~~

~~SQUADRA~~	~~SFIDA~~
~~INTEGER~~	~~INTEGER~~

~~FK SQUADRA -> SQUADRA (ID)~~

~~SFIDA -> A_SQUADRE (ID)~~

SQUADRA

ID	NOME	SFIDA -> A_SQUADRE (ID)
SERIAL	STRING	INTEGER

FA_PARTE

SPORTIVO	SQUADRA
INTEGER	INTEGER

FK SP— -> SP— (ID)

SQ— -> SQ ~~~~

T. MAX_MEMBRI, T. PARTECIPA_SQUADRE, T. DISJ_SFIDE_SQ, T. TEMPO_TAPPE

①

```
INSERT ~~/UPDATE~~ ON FA_PARTE

  \ NUMERO MEMBRI

 M = (SELECT  COUNT(*) c FROM  FA_PARTE WHERE SQUADRA = NEW.SQUADRA)

MAX = ( SELECT MAX FROM SQUADRA S, A_SQUADRE A  WHERE  S.SFIDA = A.SFIDA AND
         S.ID = NEW.SQUADRA )

IF   M.c < MAX
     COMMIT
ELSE
     GENERA ERRORE
```

②

```
INSERT ~~/UPDATE~~ ON PARTECIPA

ERROR = (EXISTS (SELECT (*) FROM A_SQUADRE A WHERE A.SFIDA = NEW.SFIDA))
IF   ERROR
       ERRORE
ELSE
       COMMIT
```

③

```
INSERT ~~/UPDATE~~ ON FA_PARTE

~~ERROR = (EXISTS ( SELECT * FROM FA_PARTE F, SQUADRA S WHERE F.SQUADRA = [illegible] S.ID )~~
~~        AND F.[illegible] S.SFIDA = NEW~~

ERROR = (EXISTS ( SELECT * FROM FA_PARTE F, SQUADRA S1, SQUADRA S2
        WHERE F.SQUADRA = S.ID AND ~~[illegible]~~ NEW.SPORTIVO = F.SPORTIVO AND
        AND S2.ID = NEW.SQUADRA AND S2.SFIDA = S1.SFIDA ))
```

(4)

~~INSERT~~

~~MK = SEL~~

~~MAX = (SELECT A.FINE FROM~~

~~MIN = (A.INIZIO~~

TMP_SPORT (U: INTEGER, S: STRING, IN: TIMESTAMP, FIN: TIMESTAMP) : REAL_GZ

IF FIN ≤ IN : ERRORE

~~RES = (SELECT A1.FINE - A1.INIZIO~~

RES = (SELECT SUM (A.FINE - A.INIZIO) FROM ALLENAMENTO A WHERE
A.UTENTE = U AND A.SPORT = S AND A.INIZIO ≥ IN AND A.FINE ≤ FIN)

RETURN RES.SUM

KM_ATT (A : INTEGER) : REAL_GZ

RES = (SELECT SUM (DB.DISTANZA (T.LONG, T.LAT, T1.LONG, T1.LAT)) FROM TAPPA T, TAPPA T1,
PERCORSO P WHERE P.ALLENAMENTO = A AND T.PERCORSO = P.ID AND T1.PERCORSO =
P.ID AND NOT EXISTS (SELECT * FROM TAPPA ~~WHERE~~ T2 WHERE T2.PERCORSO =
A AND T.ISTANTE < T2.ISTANTE AND T2.ISTANTE < T1.ISTANTE))

KM_SPORT (U:INTEGER, S:STRING, IN:TIMESTAMP, FIN:TIMESTAMP) : REAL_GZ

IF FIN < IN : ERRORE

RES = SELECT SUM (KM_ATT (A.ID)) FROM ALLENAMENTO A, PERCORSO P WHERE A.UTENTE = U AND
A.SPORT = S AND A.INIZIO ≥ IN AND A.FINE ≤ FIN AND A.ID = P.ALLENAMEN)

RESOCONTO (U: INTEGER, IN: TIMESTAMP, FIN: TIMESTAMP) : (S:STRING, REAL_GZ, REAL_GZ)

IF FIN ≤ IN : ERRORE

RES = SELECT S.NOME, SUM (A.FINE - A.INIZIO) ~~DB.TMPSPORT (U, S.NOME, IN, FIN)~~, DB.KM_SPORT (U, S.NOME, IN, FIN)
FROM SPORT S, ALLENAMENTO A WHERE A.SPORT = S.NOME AND
A.INIZIO ≥ IN AND A.FINE ≤ FIN GROUP BY S.NOME
AND A.UTENTE = U

```
TMP_AMICI (U: INTEGER, IN: TIMESTAMP, FIN: TIMESTAMP): (STR)

  IF FIN = IN: ERRORE

  ~~AMICI SELECT~~

  ~~RES = SELECT S.NOME, MAX( SUM (DB.TMP_SPORT(AM.1 , S.NOME, IN, FIN))~~
        ~~FROM AMICO A, SPORTIVO AM, SPORT S WHERE~~

  AMICI = ( SELECT U1 FROM AMICO WHERE U2 = U UNION
            U2 ~~U1~~ U1 = U);

  RES = (SELECT S.NOME, SUM (DB.TMP_SPORT(AM.U1, S.NOME, IN, FIN)) SUM FROM AMICI
        AM, SPORT S ~~WHERE AM. [illegible]~~ GROUP BY S.NOME)

  RETURN (SELECT NOME FROM RES WHERE SUM = (SELECT MAX(SUM) FROM RES))
```