

1. Requisiti sugli Utenti:

- 1.1. nome
- 1.2. cognome
- 1.3. indirizzo email
- 1.4. città di provenienza (v. req. 4)
- 1.5. data di iscrizione
- 1.6. viaggi creati come organizzatore (v. req. 2)
- 1.7. viaggi ai quali partecipa/ha partecipato (v. req. 2)
- 1.8. le attività al quale partecipa (v. req. 3)

2. Requisiti sui Viaggi:

- 2.1. nome
- 2.2. numero minimo di partecipanti
- 2.3. numero massimo di partecipanti
- 2.4. utente che lo ha organizzato (v. req. 1)
- 2.5. utenti che vi partecipano (v. req. 1)
- 2.6. attività di cui è composto

3. Requisiti sulle Attività:

- 3.1. nome
- 3.2. istante di inizio
- 3.3. durata
- 3.4. prezzo
- 3.5. indirizzo di una città dove si tiene (v. req. 4)
- 3.6. descrizione testuale
- 3.7. Se si tratta di una attività composta
- 3.8. Se è un'attività composta, le sotto-attività semplici della quale è composta
- 3.9. Se un'attività è uno spostamento
 - 3.9.1. il luogo di partenza (v. req. 4)
 - 3.9.2. il luogo di arrivo (v. req. 4)
- 3.12. Gli utenti del viaggio che partecipano all'attività (v. req. 1)

4. Requisiti sui Luoghi:

- 4.1. indirizzo
- 4.2. città
- 4.3. regione
- 4.4. nazione

999.1. il sistema deve permettere di specificare, per ogni attività inclusa in un viaggio, gli utenti che vi partecipano.

999.2. Il sistema deve permettere agli utenti registrati di partecipare a viaggi e di creare viaggi in qualità di organizzatori, aggiungendo a essi attività di vario tipo.

999.3. trovare tutti i viaggi che includono una certa destinazione in un certo intervallo di tempo.

999.4. trovare la/e città toccata/e dal maggior numero di viaggi in un certo intervallo di tempo;

999.5. calcolare, per ogni regione di una data nazione, il numero di viaggi organizzati in un dato periodo di tempo che toccano tale regione;

999.6. dato un budget minimo, un budget massimo, un insieme di regioni, un periodo di

tempo e un numero N , calcolare l'insieme dei viaggi che hanno un budget nell'intervallo richiesto, toccano almeno una delle regioni date, si svolgono interamente all'interno del periodo dato e sono organizzati da un utente che abbia organizzato almeno N viaggi in precedenza.

999.7. I responsabili del sistema devono poter calcolare, data una città di destinazione, per ognuno dei 12 mesi dell'anno, il numero di viaggi organizzati in quel mese nell'ultimo anno solare.

(Solo e unicamente per le funzionalità di seguito, dare la specifica completa.)

999.8. Dato un viaggio e un utente che vi partecipa, calcolare il numero di attività del viaggio di tipo "spostamento" alle quali ha preso parte.

999.9. Dato un viaggio, ottenere gli utenti che hanno partecipato al viaggio e che hanno preso parte al maggior numero di attività organizzate in esso.

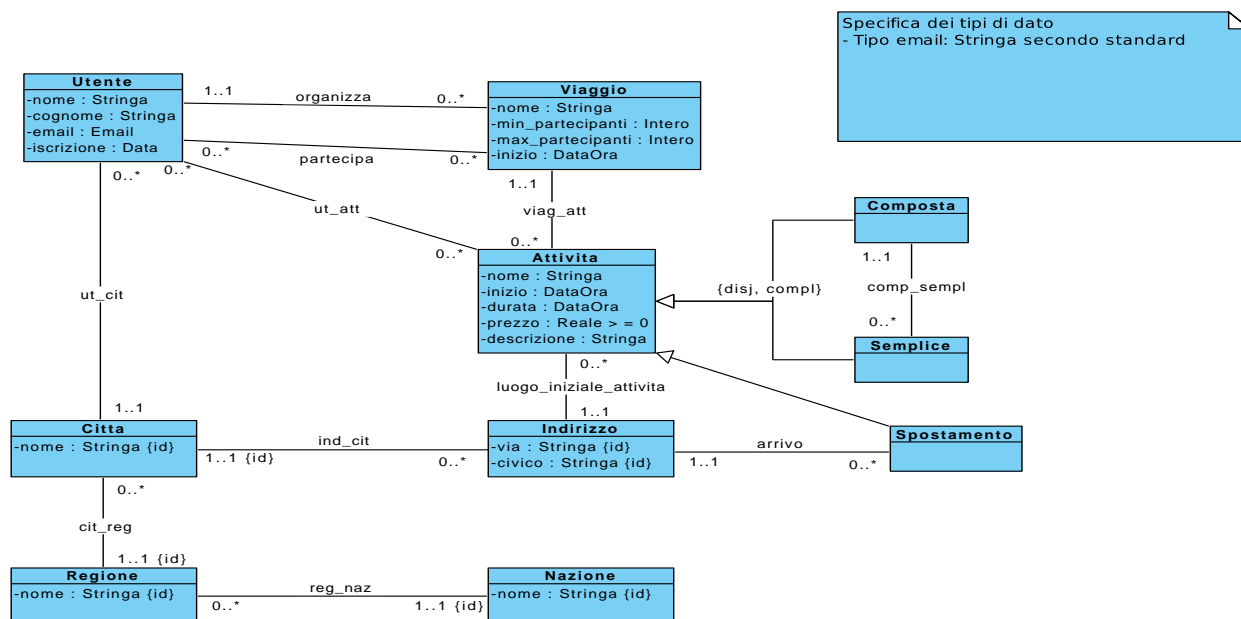
999.10. I responsabili del sistema, data una nazione, devono poter calcolare, per ognuna delle sue regioni, la/e città toccata/e dal maggior numero di viaggi.

999.11. Data un'attività, calcolare il numero di utenti che vi ha preso parte.

999.12. Dato un viaggio, ottenere la/e attività organizzata/e in esso alla/e quale/i ha preso parte il maggior numero di utenti.

999.13. Data una città di destinazione, per ognuno dei 12 mesi dell'anno, calcolare il numero di viaggi organizzati in tale mese nell'ultimo anno solare, in tale città

sd



Funzioni ausiliarie

sovrapposti($i1 : \text{DataOra}$, $i2 : \text{DataOra}$, $f1 : \text{DataOra}$, $f2 : \text{DataOra}$) : Booleano:

- precondizioni:
 - Deve essere vero che $i1 \leq f1$ e che $i2 \leq f2$
- postcondizioni:
 - result = true se $i1 \leq f2$ e $i2 \leq f1$ altrimenti result = false

Specifica della classe Utente

[V.Utente.partecipa_ad_attivita_prima_iscrizione] Un utente non può partecipare ad attività che sono iniziate prima della sua iscrizione

Per ogni $ut: \text{Utente}$, sia $att: \text{Attivita}$ tale che $(ut, att):ut_att$, deve essere vero che $ut.iscrizione \leq att.inizio$

Specifica della classe Viaggio

[V.Viaggio.date_valide] La data di inizio deve venire prima della data di fine

Per ogni $viag: \text{Viaggio}$, deve essere vero che $viag.inizio \leq viag.fine$

[V.Viaggio.numero_corretto_partecipanti] Il numero di utenti che partecipano ad un viaggio deve essere coerente con i valori dei relativi attributi

Per ogni $viag: \text{Viaggio}$, sia U l'insieme di tutti gli $ut: \text{Utente}$ che partecipano nel link dell'associazione $(ut, viag):partecipa$.

Deve essere vero che $viag.min_partecipanti \leq |U| \leq viag.max_partecipanti$

Specifiche della classe Attivita

$fine() : \text{DataOra}$:

- precondizioni:
 - nessuna
- postcondizioni:
 - L'operazione non modifica il livello estensionale
 - Sia result = $this.inizio + this.durata$

[V.Attivita.data_attivita_valida] La data di inizio dell'attività deve essere dopo la data di inizio del viaggio e la data di fine dell'attività deve essere prima della data di fine del viaggio

Per ogni $att: \text{Attivita}$, sia $viag: \text{Viaggio}$ tale che $(viag, att):viag_att$ deve essere vero che $viag.inizio \leq att.inizio \leq att.fine() \leq viag.fine$

[V.Attivita.utenti_partecipano_anche_al_viaggio] Tutti gli utenti che partecipano all'attività X devono anche partecipare al viaggio di quella attività X

Per ogni $att: \text{Attivita}$:

- sia $U1$ l'insieme di tutti gli $ut: \text{Utente}$ tali che $(ut, att):ut_att$.
- sia $viag: \text{Viaggio}$ tale che $(viag, att):viag_att$,
 - sia $U2$ l'insieme di tutti gli $ut: \text{Utente}$ tali che $(ut, viag):partecipa$

Deve essere vero che ogni $ut: \text{Utente}$ in $U1$ è anche presente in $U2$

[V.Attivita.utente_non_partecipare_due_attivita_contemporaneamente] Un utente non può partecipare a due attività che si sovrappongono temporalmente.

Per ogni $ut: \text{Utente}$, sia $att': \text{Attivita}$ e $att'': \text{Attivita}$,
tali che $att' \neq att''$, che $(ut, att'):ut_att$ e che $(ut, att''):ut_att$

non deve essere vero che sovrapposti(att'.inizio, att".inizio, att'.fine(), att".fine())

numero_spostamenti(viag: Viaggio, ut: Utente): Intero:

- preconditioni:
 - deve essere vero che (ut, viag):partecipa
- postcondizioni:
 - Sia S l'insieme di tutti gli sp:Spostamento, tali che (viag, sp):viag_att e che (ut, sp):ut_att
 - result = |S|

utenti_con_max_attivita(viag: Viaggio) : Utente[0..*]:

- preconditioni:
 - nessuna
- postcondizioni:
 - Sia U l'insieme delle coppie (ut, n), dove:
 - ut:Utente, tale che esiste il link (ut, viag):partecipa
 - n è il numero di att:Attivita, tali che:
 - Esiste il link (ut, att):ut_att
 - Esiste il link (viag, att):viag_att
 - Sia U_Max l'insieme degli elementi (ut, n) di U che hanno n massimo
 - result è l'insieme formato da tutti gli elementi ut dell'insieme U_Max

citta_piu_viaggiate(naz: Nazione) :

- preconditioni:
 - nessuna
- postcondizioni:
 - Sia Res l'insieme di tutte le coppie (reg, C_Max), dove:
 - reg:Regione, tale che esista il link (reg, naz):reg_naz
 - C_Max è definito come segue:
 - Sia C l'insieme di tutte le coppie (cit, n2), dove:
 - cit:Citta, tale che esista il link (cit, reg):cit_reg
 - n2 è il numero di tutte i viag:Viaggio, tali che:
 - Sia A l'insieme di tutte le att:Attivita tali che (viag, att):viag_att,
 - e sia I l'insieme di tutti gli ind:Indirizzo tali che (ind, cit):ind_cit,
 - esista almeno una coppia di att e ind che insieme partecipano al link dell'associazione

luogo_iniziale_attivita

- C_Max è l'insieme di tutti gli elementi cit per i quali il valore di n2 è massimo
- result = Res

citta_con_piu_viaggi(naz:Nazione) :

- preconditioni:
 - nessuna
- postcondizioni:
 - Sia

utenti_in_attivita(att: Attivita) : Intero:

- Sia U l'insieme di tutti gli ut:Utente tali che (ut, att):ut_att
- result = |U|

attivit _con_max_utenti(viag: Viaggio) : Attivit [0..*]

- precondizioni:

- nessuna

- postcondizioni:

- Sia A l'insieme formato dalle coppie (att, n), dove:

- att:Attivit , tale che esiste il link (viag, att):viag_att

- n   il numero di ut:Utente tali che esiste il link (ut, att):ut_att

- Sia A_Max l'insieme degli elementi (att, n) di A che hanno n massimo

- result   l'insieme formato da tutti gli elementi att dell'insieme A_Max

numero_viaggi_per_mese(cit:Citt ) :

- precondizioni:

- nessuna

- postcondizioni

- Sia V l'insieme di 12 coppie (m, n), dove:

- m corrisponde a ciascuno dei 12 mesi dell'anno solare (Gennaio, Febbraio, Marzo, ..., Dicembre)

- n   il numero di viag:Viaggio tali che:

- il mese di viag.inizio = m

- l'anno di viag.inizio = l'anno di adesso

- Sia A l'insieme di tutte le att:Attivit  tali che (viag, att):viag_att,

e sia I l'insieme di tutti gli ind:Indirizzo tali che (ind, cit):ind_cit,

esista almeno una coppia di att e ind che insieme partecipano al link dell'associazione

luogo_iniziale_attivit  - result   l'insieme V

