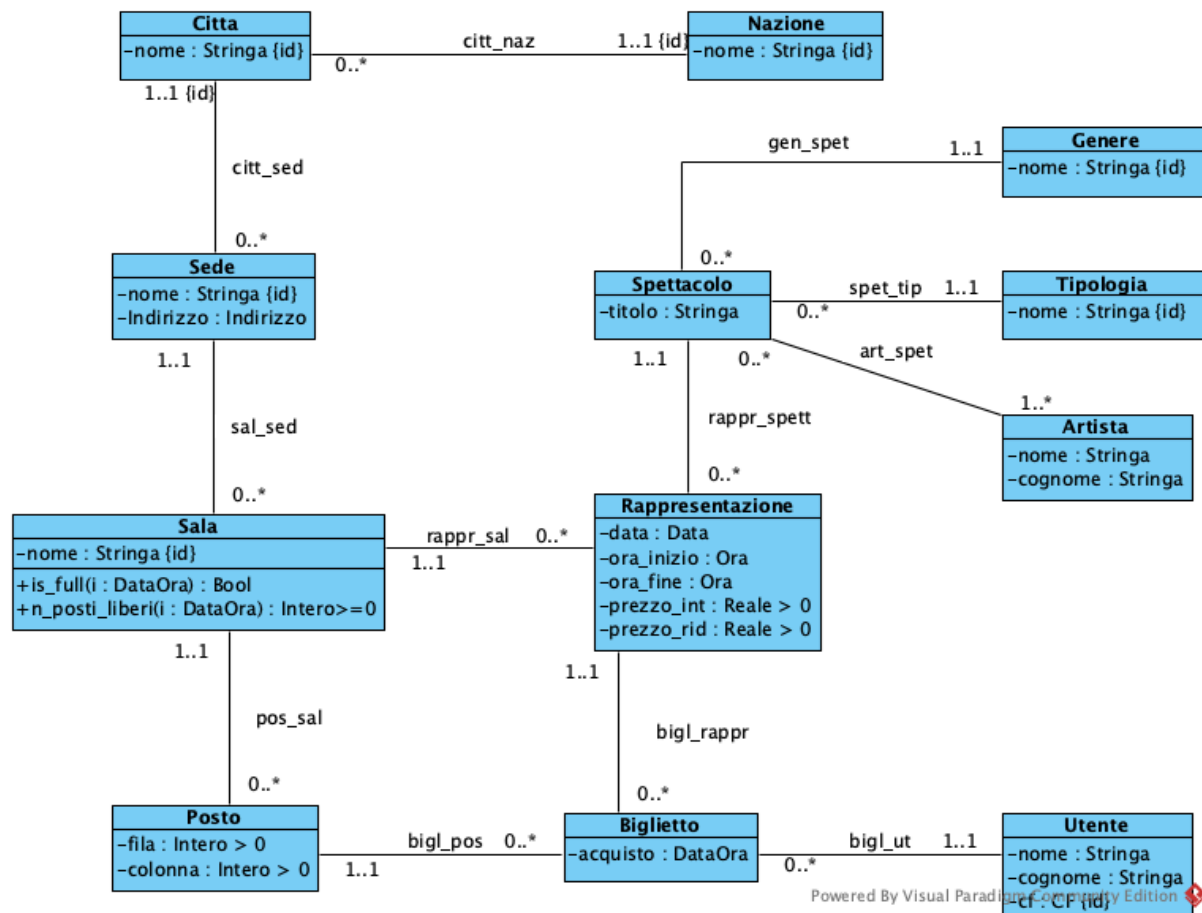


# Out!

## Raffinamento dei requisiti

1. Delle sedi interessa:
    - 1.1. Nome
    - 1.2. Indirizzo
    - 1.3. Sale (v.req.2)
  2. Delle sale interessa:
    - 2.1. Spettacolo che ospita (v.req. 3)
    - 2.2. Posti a sedere, di cui interessa:
      - 2.2.1. Numero di fila
      - 2.2.2. Colonna
      - 2.2.3. Il prezzo (pieno/ridotto), in base allo spettacolo
  3. Degli spettacoli interessa:
    - 3.1. Il titolo
    - 3.2. La tipologia (concerto, rappresentazione teatrale, film)
    - 3.3. Il genere
    - 3.4. Gli artisti
    - 3.5. Le date
    - 3.6. Gli orari
    - 3.7. LE sedi (v.req. 1)
    - 3.8. Le sale (v.req. 2)
  4. Degli utenti interessa:
    - 4.1. Spettacoli a cui hanno partecipato:
      - 4.1.1. Posto a sedere (v.req. 2.2.)
      - 4.1.2. Sede (v.req. 1)
999. Il sistema deve offrire le seguenti funzionalità:
- 999.1. Iscrivere al servizio come **utente**.
  - 999.2. Prenotare uno o più posti a prezzo pieno o ridotto di un dato settore per una data di uno spettacolo. I posti non sono liberi, pertanto il sistema deve assegnare, all'atto della prenotazione, un numero di posti sufficienti non ancora prenotati.
  - 999.3. Consultare la lista degli spettacoli di una certa tipologia e genere (ad es., spettacoli teatrali/commedie) previsti in un certo giorno.
  - 999.4. Ricevere dal sistema suggerimenti di nuovi spettacoli da vedere. In particolare, si richiede che il sistema segnali ad un utente l'insieme degli spettacoli programmati nei successivi 7 giorni, che sono dello stesso genere (anche se di tipologia diversa) dell'ultimo spettacolo che egli ha prenotato.

# Diagramma UML delle classi



## Tipi di dato

Indirizzo: {  
 via: Stringa  
 civico: stringa numerica  
 cap: CAP  
 }

CAP: stringa secondo standard  
 CF: stringa secondo standard

## Specifica della classe Sala

n\_posti\_liberi(i:DataOra): Intero >= 0  
 precondizioni:

postcondizioni:

is\_full(i: DataOra): Bool

precondizioni:

postcondizioni:

## Specifica dei vincoli esterni

### [V.SalaRappresentazione.nonSovrapposti]

Una sala non può avere più spettacoli sovrapposti

Per ogni  $s:Sala$ ,  $p1:Rappresentazione$  e  $p2:Rappresentazione$  tali che  $p1 \neq p2$  e  $p1.data == p2.data$  e  $(p1,s):rapp_rsal$  e  $(p2,s):rapp_rsal$  allora  $p1.ora\_inizio > p2.ora\_fine$  e  $p1.ora\_fine < p2.ora\_inizio$ .

### [V.BigliettoPostoRappresentazione.stessaSala]

Dato un biglietto per una certa rappresentazione e un certo posto, il posto e la rappresentazione devono riferirsi alla stessa sala.

Per ogni  $b:Biglietto$  t.c.  $(b,p):bigl\_pos$  e  $(b,r):bigl\_rapp_r$ , allora deve essere  $(p,s):pos\_sal$  e  $(r,s):rapp\_sal$ .

### [V.BigliettoPostoRappresentazione.nonDoppiaVendita]

Un biglietto non può essere venduto più di una volta per lo stesso posto durante la stessa rappresentazione.

Non esiste  $b1:Biglietto$  e  $b2:Biglietto$  t.c.  $b1 \neq b2$  e  $(b1,p):bigl\_pos$  e  $(b2,p):bigl\_pos$  e  $(b1,r):bigl\_rapp_r$  e  $(b2,r):bigl\_rapp_r$ .

### [V.BigliettoSalaRappresentazione.capacitaMax]

Non si possono vendere più biglietti di quanti sono i posti disponibili per una certa rappresentazione.

Per ogni  $i:DataOra$  deve essere che non esiste  $b:Biglietto$  t.c.  $(b,p):bigl\_pos$ ,  $(p,s):pos\_sal$ ,  $b.acquisto > i$  e  $s.n\_posti\_liberi(i) = 0$ .

# Specifica delle operazioni di use case

iscrivi\_utente(n:String, c:String, cf:CF): Utente

precondizioni:

postcondizioni:

... per esercizio

prenota\_posti(r:Rappresentazione, p:Posto): Biglietto

precondizioni:

postcondizioni:

...

...