

# Progettazione e implementazione di una base di dati relazionale per la gestione di conferenze scientifiche

Caporaso Antonio Di Fusco Giorgio

N86003458

N86004389

23 giugno 2023

## Indice

<b>1</b>	<b>Traccia</b>	<b>2</b>
1.1	Output attesi dal committente . . . . .	2
<b>2</b>	<b>Progettazione concettuale</b>	<b>2</b>
2.1	Analisi dei dati . . . . .	2
<b>3</b>	<b>Schema concettuale</b>	<b>2</b>
<b>4</b>	<b>Ristrutturazione dello schema concettuale</b>	<b>3</b>
4.1	Rimozione degli attributi multivalore . . . . .	3
4.2	Rimozione classi di associazione . . . . .	4
4.3	Rimozione generalizzazioni . . . . .	4
4.4	Scelta degli identificatori principali . . . . .	4
<b>5</b>	<b>Progettazione logica</b>	<b>4</b>
5.1	Traduzione delle classi . . . . .	4
5.2	Traduzione delle associazioni . . . . .	5
5.2.1	Traduzione delle associazioni molti a molti . . . . .	5
5.2.2	Traduzione delle associazioni uno a molti . . . . .	6
5.2.3	Traduzione delle associazioni uno a uno . . . . .	7
5.3	Schema logico . . . . .	7
<b>6</b>	<b>Implementazione fisica</b>	<b>7</b>
6.1	Definizione delle tabelle . . . . .	7
6.2	Popolamento . . . . .	10
6.3	Trigger . . . . .	10
6.4	Funzioni . . . . .	10
6.5	Procedure . . . . .	10
<b>A</b>	<b>Dizionario dei dati</b>	<b>10</b>
<b>B</b>	<b>Dizionario delle associazioni</b>	<b>10</b>
<b>C</b>	<b>Dizionario dei vincoli</b>	<b>10</b>

# 1 Traccia

Si sviluppi un sistema informativo, composto da una base di dati relazionale e da un applicativo Java dotato di GUI (Swing o JavaFX), per la gestione di **conferenze scientifiche**.

Ogni conferenza ha una data di inizio e di fine, una collocazione (sede, indirizzo), uno o più enti che la organizzano, degli sponsor (che coprono in parte le spese), una descrizione, ed un gruppo di organizzatori, che può essere distinto in comitato scientifico e comitato locale (che si occupa cioè della logistica). Di ognuno degli organizzatori, così come di tutti i partecipanti, si riportano titolo, nome, cognome, email ed istituzione di appartenenza.

Ogni conferenza può avere una o più sessioni, anche in parallelo fra loro. Ogni sessione ha una locazione all'interno della sede. Per ogni sessione c'è un programma, che prevede la presenza di un coordinatore (chair) che gestisce la sessione, ed eventualmente di un keynote speaker (un partecipante di particolare rilievo invitato dagli organizzatori). Ogni sessione avrà quindi una successione di interventi ad orari predefiniti e di specifici partecipanti. Per ogni intervento si conserva un abstract (un breve testo in cui viene spiegato il contenuto del lavoro presentato).

Si deve poter considerare la presenza di spazi di intervallo (coffee breaks, pranzo) ma anche la presenza di eventi sociali (cene, gite, etc).

## 1.1 Output attesi dal committente

1. Documento di Design della base di dati:
  - (a) Class Diagram della base di dati.
  - (b) Dizionario delle Classi, delle Associazioni e dei Vincoli.
  - (c) Schema Logico con descrizione di Trigger e Procedure individuate.
2. File SQL contenenti:
  - (a) Creazione della struttura della base di dati.
  - (b) Popolamento del DB.
  - (c) (Facoltativo, ma apprezzato) README contenente i commenti all'SQL.

# 2 Progettazione concettuale

## 2.1 Analisi dei dati

Le entità che possono essere individuate nel problema sono elencate all'interno della Tabella 1.

# 3 Schema concettuale

Nella Figura 1 è presente lo schema concettuale della base di dati descritta nella sezione 1.

Entità	Descrizione
<b>Conferenza</b>	Per le conferenze delle quali si vuole poter gestire le informazioni. Di ogni conferenza si conservano il <i>nome</i> , l' <i>inizio</i> e la <i>fine</i> e una <i>descrizione</i> .
<b>Ente</b>	Per gli enti che organizzano le conferenze scientifiche. Di ogni ente si conserva il <i>nome</i> e la <i>sigla</i> .
<b>Sponsor</b>	Per gli sponsor che coprono le spese della conferenza. Di ogni sponsor si conserva il <i>nome</i> .
<b>Comitato</b>	Per i gruppi di organizzatori che si occupano della gestione della conferenza. Si distinguono in comitati <i>scientifici</i> e <i>locali</i> .
<b>Organizzatore</b>	Per i membri dei comitati. Di ogni organizzatore si riportano <i>titolo</i> , <i>nome</i> , <i>cognome</i> , <i>email</i> ed <i>istituzione di appartenenza</i> .
<b>Sede</b>	Per descrivere il luogo dove si tengono le varie conferenze. Di ogni sede si conservano il <i>nome</i> , l' <i>indirizzo</i> e la <i>città</i> .
<b>Sala</b>	Per tenere traccia dell'ubicazione delle varie sessioni. Di ogni sala si conserva il <i>nome della sala</i> e la sua <i>capacità</i> .
<b>Sessione</b>	Per rappresentare le sessioni di una conferenza. Per ogni sessione si riporta il <i>titolo</i> , un <i>coordinatore</i> , data e orario d' <i>inizio</i> e di <i>fine</i> .
<b>Programma</b>	Per il programma di ciascuna sessione. Ogni programma specifica la presenza di un <i>keynote speaker</i> , ovvero un partecipante di rilievo.
<b>Intervento</b>	Per i vari interventi di una sessione. Per ogni intervento si conserva un <i>abstract</i> , il partecipante ( <i>speaker</i> ) che effettua l'intervento e l' <i>orario</i> dello stesso.
<b>Partecipante</b>	Per i partecipanti delle varie sessioni. Ogni partecipante ha gli stessi attributi degli organizzatori.
<b>Intervallo</b>	Per descrivere i vari intervalli presenti all'interno di una sessione. Questi possono essere di due tipologie: <i>coffee break</i> oppure dei <i>pranzi</i> . Per ogni intervallo si riporta l' <i>orario</i> .
<b>Evento sociale</b>	Per i vari eventi sociali previsti all'interno di una sessione. Questi possono essere di varia natura. Come per gli intervalli se ne riporta l' <i>orario</i> .

Tabella 1: Entità del problema

## 4 Ristrutturazione dello schema concettuale

### 4.1 Rimozione degli attributi multivalore

All'interno del diagramma delle classi mostrato in Figura 1 sono presenti vari attributi multivalore. Per ciascuno di essi sono state fatte le seguenti valutazioni:

1. Si partiziona l'attributo *Indirizzo* di SEDE suddividendolo in vari campi *Via*, *Civico* e *Cap*.
2. Si è deciso di partizionare l'attributo *Valuta* presente nella classe di associazione SPONSORIZZAZIONE creando una nuova classe chiamata VALUTA.

## 4.2 Rimozione classi di associazione

All'interno dello schema concettuale è presente la classe di associazione SPONSORIZZAZIONE all'interno dell'associazione [...] tra CONFERENZA e SPONSOR. Nello schema ristrutturato questa è stata rimossa reificandola e scindendo l'associazione in due associazioni di tipo [1..\*].

## 4.3 Rimozione generalizzazioni

Per quanto riguarda la rimozione delle generalizzazioni presenti nello schema concettuale:

1. Nel caso delle entità COMITATO SCIENTIFICO e COMITATO LOCALE che specializzano la classe COMITATO si è optato per l'accorpamento delle classi figlie all'interno della super-classe attraverso la specifica di una enumerazione chiamata COMITATO\_ST composta dai campi *Scientifico* e *Locale*;
2. Nel caso delle entità PRANZO e COFFEE BREAK che specializzano la classe INTERVALLO si è adottato la stessa politica.

## 4.4 Scelta degli identificatori principali

Risulta conveniente ai fini di una migliore traduzione delle associazioni l'introduzione di chiavi surrogate per ogni entità. Tali chiavi altro non saranno che identificativi numerici interi del tipo *Id\_NomeEntità*, eccezion fatta per l'entità VALUTA la quale viene identificata univocamente da una stringa di tre caratteri stando allo standard ISO 4217<sup>1</sup>.

# 5 Progettazione logica

Una volta aver ristrutturato lo schema concettuale mostrato in Figura 1 si procede traducendo le varie associazioni descritte in Figura 2. Iniziamo col tradurre direttamente tutte le classi. Man mano che si andranno a tradurre le varie associazioni andremo a modificare la struttura dei vari schemi relazionali laddove necessario.

## 5.1 Traduzione delle classi

Si ha quindi:

ENTE

<u>id_ente</u>	nome	sigla
----------------	------	-------

SEDE

<u>id_sede</u>	nome	via	civico	cap	city
----------------	------	-----	--------	-----	------

SPONSOR

<u>id_Sponsor</u>	Nome
-------------------	------

COMITATO

<u>id_Comitato</u>	Tipologia
--------------------	-----------

ORGANIZZATORE

<u>id_Organizzatore</u>	Nome	Cognome	Titolo	Email
-------------------------	------	---------	--------	-------

---

<sup>1</sup>ISO 4217 è uno standard internazionale che descrive codici di tre lettere per definire i nomi delle valute, stabilito dall'Organizzazione internazionale per la normazione (ISO), che viene usato comunemente nel sistema bancario e nel mondo economico, nonché nella stampa specializzata.

SALA

<u>id_Sala</u>	Nome	Capienza
----------------	------	----------

CONFERENZA

<u>id_Conferenza</u>	Nome	Descrizione	Inizio	Fine
----------------------	------	-------------	--------	------

PARTECIPANTE

<u>id_Partecipante</u>	Nome	Cognome	Titolo	Email
------------------------	------	---------	--------	-------

SESSIONE

<u>id_Sessione</u>	Nome	Inizio	Fine
--------------------	------	--------	------

VALUTA

<u>Iso</u>	Nome	Simbolo
------------	------	---------

SPEAKER

<u>id_Speaker</u>	Nome	Cognome	Titolo	Email
-------------------	------	---------	--------	-------

PROGRAMMA

<u>Id_Programma</u>
---------------------

INTERVALLO

<u>id_Intervallo</u>	Tipologia	Inizio	Fine
----------------------	-----------	--------	------

INTERVENTO

<u>id_Intervento</u>	Titolo	Abstract	Inizio	Fine
----------------------	--------	----------	--------	------

EVENTO

<u>id_Evento</u>	Tipologia	Inizio	Fine
------------------	-----------	--------	------

## 5.2 Traduzione delle associazioni

### 5.2.1 Traduzione delle associazioni molti a molti

Traduciamo le associazioni \*.\* mediante la realizzazioni di apposite tabelle ponte. Si ha allora:

1. L'associazione ENTECONFERENZA tra ENTE e CONFERENZA:

ENTECONFERENZA

<u>id_ente</u>	<u>id_conferenza</u>
----------------	----------------------

2. L'associazione ORGANIZZATORECOMITATO tra ORGANIZZATORE e COMITATO:

ORGANIZZATORECOMITATO

<u>id_organizzatore</u>	<u>id_comitato</u>
-------------------------	--------------------

3. L'associazione PARTECIPANTESESSIONE tra PARTECIPANTE e SESSIONE:

PARTECIPANTESESSIONE

<u>id_Partecipante</u>	<u>id_Sessione</u>
------------------------	--------------------

### 5.2.2 Traduzione delle associazioni uno a molti

Per ciascuna delle associazioni binarie di tipo uno a molti si identificano le entità deboli e quelle forti che partecipano all'associazione. Per tradurre l'associazione in relazioni basterà includere la chiave surrogata dell'entità forte all'interno della relazione dell'entità debole. Avremo quindi:

1. Associazioni di composizione:

- (a) Una sede è composta da più sale quindi:

SALA

<u>id_Sala</u>	Nome	Capienza	<u>id_sede</u>
----------------	------	----------	----------------

- (b) Una conferenza è composta da più sessioni:

SESSIONE

<u>id_Sessione</u>	Nome	Inizio	Fine	<u>id_conferenza</u>
--------------------	------	--------	------	----------------------

- (c) Un programma è composto da interventi, intervalli ed eventi:

INTERVALLO

<u>id_Intervallo</u>	Tipologia	Inizio	Fine	<u>id_programma</u>
----------------------	-----------	--------	------	---------------------

INTERVENTO

<u>id_Intervento</u>	Titolo	Abstract	Inizio	Fine	<u>id_programma</u>
----------------------	--------	----------	--------	------	---------------------

EVENTO

<u>id_Evento</u>	Tipologia	Inizio	Fine	<u>id_programma</u>
------------------	-----------	--------	------	---------------------

2. Un partecipante, uno speaker ed un organizzatore appartengono ad una istituzione, ovvero un ENTE:

SPEAKER

<u>id_Speaker</u>	Nome	Cognome	Titolo	Email	<u>id_ente</u>
-------------------	------	---------	--------	-------	----------------

PARTECIPANTE

<u>id_Speaker</u>	Nome	Cognome	Titolo	Email	<u>id_ente</u>
-------------------	------	---------	--------	-------	----------------

ORGANIZZATORE

<u>id_Speaker</u>	Nome	Cognome	Titolo	Email	<u>id_ente</u>
-------------------	------	---------	--------	-------	----------------

3. Ogni intervento ha uno speaker che lo effettua:

INTERVENTO

<u>id_Intervento</u>	<u>id_speaker</u>	Titolo	Abstract	Inizio	Fine	<u>id_programma</u>
----------------------	-------------------	--------	----------	--------	------	---------------------

4. Una sala può ospitare più sessioni:

SESSIONE

<u>id_Sessione</u>	Nome	Inizio	Fine	<u>id_sala</u>	<u>id_conferenza</u>
--------------------	------	--------	------	----------------	----------------------

5. Una sede può ospitare più conferenze:

CONFERENZA

<u>id_Conferenza</u>	Nome	Descrizione	Inizio	Fine	<u>id_sede</u>
----------------------	------	-------------	--------	------	----------------

- Una conferenza ha due comitati, uno scientifico ed uno locale:

CONFERENZA

<u>id_Conferenza</u>	Nome	Descrizione	Inizio	Fine	<u>id_sede</u>	<u>id_com_locale</u>	<u>id_com_scientifico</u>
----------------------	------	-------------	--------	------	----------------	----------------------	---------------------------

### 5.2.3 Traduzione delle associazioni uno a uno

Si ha:

- Ogni sessione ha un coordinatore:

SESSIONE

<u>id_Sessione</u>	Nome	Inizio	Fine	<u>id_sala</u>	<u>id_conferenza</u>	<u>id_coordinatore</u>
--------------------	------	--------	------	----------------	----------------------	------------------------

- Ogni programma si riferisce ad una sessione e ad un keynote speaker:

PROGRAMMA

<u>Id_Programma</u>	<u>id_Sessione</u>	<u>id_keynote</u>
---------------------	--------------------	-------------------

## 5.3 Schema logico

Nella Figura 3 è raffigurato lo schema logico risultante.

# 6 Implementazione fisica

## 6.1 Definizione delle tabelle

```

1 CREATE SCHEMA conference;
2 set search_path to conference;
3
4 --Definizione tipi
5 create type comitato_st as enum ('locale','scientifico');
6 create type intervallo_st as enum ('pranzo','coffee break');
7
8 --Tabella delle istituzioni
9 create table ente(
10 id_ente serial primary key,
11 nome text not null unique,
12 sigla varchar(7) not null,
13 unique (nome,sigla)
14 );
15
16 --Tabella delle sedi
17 create table sede(
18 id_serial serial primary key,
19 nome text not null,
20 via text not null,
21 civico varchar(5) not null,
22 cap varchar(5) not null,
23 city text not null
24 );
25
26 -- Tabella degli sponsor
27 create table sponsor(
28 id_sponsor serial primary key,
29 nome text not null

```

```

30 );
31
32 -- Tabella dei comitati
33 create table comitato(
34 id_comitato serial primary key,
35 tipologia comitato_st not null
36 );
37
38 -- Tabella degli organizzatori
39 create table organizzatore(
40 id_organizzatore serial primary key,
41 nome text not null,
42 cognome text not null,
43 titolo varchar(10) not null,
44 email text not null unique,
45 id_ente integer references ente(id_ente) on delete cascade
46 );
47
48 -- Tabella delle sale
49 create table sala(
50 id_sala serial primary key,
51 nome text not null,
52 capienza integer not null,
53 id_sede integer references sede(id_serial) on delete cascade
54 );
55
56 -- Tabella delle conferenze
57 create table conferenza(
58 id_conferenza serial primary key,
59 titolo text not null,
60 descrizione text not null,
61 inizio timestamp not null,
62 fine timestamp not null,
63 id_sede integer references sede(id_serial) on delete set null,
64 id_comitato_scientifico integer references comitato(id_comitato) on delete
    set null,
65 id_comitato_locale integer references comitato(id_comitato) on delete set
    null,
66 check (inizio <= fine),
67 check (inizio >= now())
68 );
69
70 -- Tabella dei partecipanti
71 create table partecipante(
72 id_partecipante serial primary key,
73 nome text not null,
74 cognome text not null,
75 titolo varchar(10) not null,
76 email text not null unique,
77 id_ente integer references ente(id_ente) on delete set null
78 );
79
80 -- Tabella delle sessioni
81 create table sessione(
82 id_sessione serial primary key,
83 titolo text not null,
84 inizio timestamp not null,
85 fine timestamp not null,
86 id_coordinatore integer references organizzatore(id_organizzatore) on delete
    set null,

```



```

87 id_conferenza integer references conferenza(id_conferenza) on delete cascade,
88 id_sala integer references sala(id_sala) on delete set null,
89 check (inizio <= fine)
90 );
91
92 --Tabella ponte tra la tabella Sessione e la tabella Partecipante
93 create table PartecipanteSessione(
94 id_partecipante integer references partecipante(id_partecipante) on delete
95 cascade,
96 id_sessione integer references sessione(id_sessione) on delete cascade,
97 unique (id_partecipante,id_sessione)
98 );
99
100 -- Tabella ponte tra le tabelle Ente e la tabella Conferenza
101 create table ente_conferenza(
102 id_ente integer references ente(id_ente) on delete cascade,
103 id_conferenza integer references conferenza(id_conferenza) on delete cascade,
104 unique (id_ente,id_conferenza)
105 );
106
107 -- Tabella per la rappresentazione delle valute
108 create table valuta(
109 iso char(3) primary key,
110 nome text not null,
111 simbolo text not null
112 );
113
114 --Tabella ponte tra Sponsor e Conferenza
115 create table SponsorConferenza(
116 id_sponsor integer references sponsor(id_sponsor) on delete cascade,
117 contributo numeric(1000,2) not null,
118 valuta char(3) references valuta(iso) not null,
119 id_conferenza integer references conferenza(id_conferenza) on delete cascade,
120 unique (id_sponsor,id_conferenza)
121 );
122
123 -- Tabella per gli speaker delle sessioni
124 create table speaker(
125 id_speaker serial primary key,
126 nome text not null,
127 cognome text not null,
128 titolo varchar(10) not null,
129 email text not null unique,
130 id_ente integer references ente(id_ente) on delete set null
131 );
132
133 -- Tabella per i programmi delle sessioni
134 create table programma(
135 id_programma serial primary key,
136 id_sessione integer references sessione(id_sessione) on delete cascade,
137 id_keynote integer references speaker(id_speaker) on delete set null,
138 unique (id_programma, id_sessione)
139 );
140
141 -- Tabella per gli interventi in programma
142 create table intervento(
143 id_intervento serial primary key,
144 titolo text not null,
145 abstract text not null,
146 inizio timestamp not null,

```

```

146 fine timestamp not null,
147 id_speaker integer references speaker(id_speaker) on delete cascade,
148 id_programma integer references programma(id_programma) on delete cascade,
149 unique (id_speaker,id_programma),
150 check (inizio <= fine)
151 );
152
153 -- Tabella per gli intervalli in programma
154 create table intervallo(
155 id_intervallo serial primary key,
156 tipologia intervallo_st not null,
157 inizio timestamp not null,
158 fine timestamp not null,
159 check (inizio <= fine),
160 id_programma integer references programma(id_programma) on delete cascade
161 );
162
163 -- Tabella per gli eventi in programma
164 create table evento(
165 id_evento serial primary key,
166 tipologia text not null,
167 inizio timestamp not null,
168 fine timestamp not null,
169 check (inizio <= fine),
170 id_programma integer references programma(id_programma) on delete cascade
171 );
172
173 -- Tabella ponte tra Organizzatore e Comitato
174 create table organizzatore_comitato(
175 id_organizzatore_comitato serial primary key,
176 id_organizzatore integer references organizzatore(id_organizzatore) on delete
    cascade,
177 id_comitato integer references comitato(id_comitato) on delete cascade,
178 unique (id_organizzatore,id_comitato)
179 );
180

```

## 6.2 Popolamento

## 6.3 Trigger

## 6.4 Funzioni

## 6.5 Procedure

## A Dizionario dei dati

## B Dizionario delle associazioni

## C Dizionario dei vincoli

## Elenco delle figure

1	Schema concettuale del problema . . . . .	12
2	Ristrutturazione dello schema concettuale . . . . .	13
3	Schema logico . . . . .	14

## Elenco delle tabelle

1	Entità del problema . . . . .	3
---	-------------------------------	---

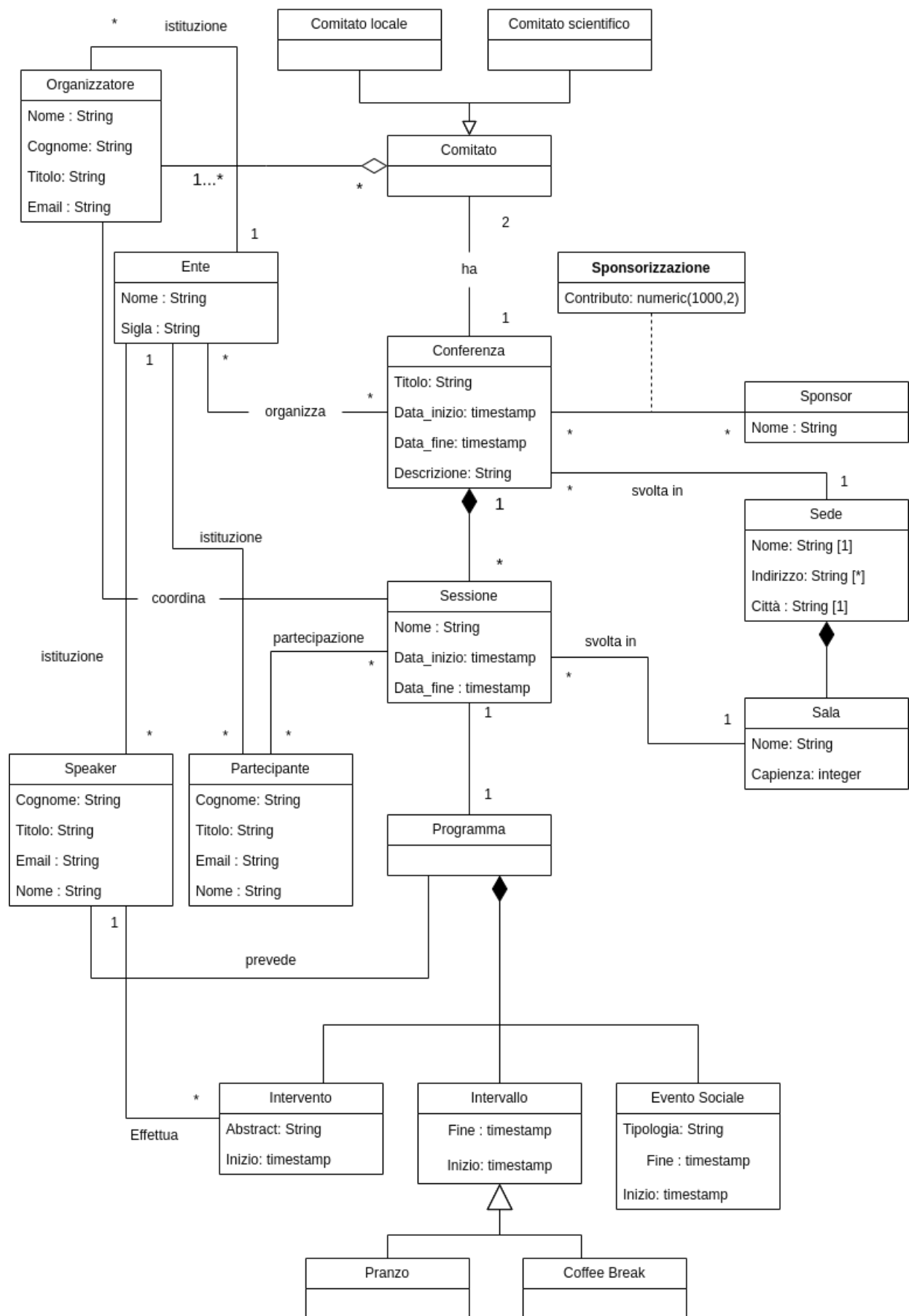


Figura 1: Schema concettuale del problema



Figura 3: Schema logico

