

UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO II
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA ELETTRICA E TECNOLOGIE DELL'INFORMAZIONE



CORSO DI LAUREA IN INFORMATICA
INSEGNAMENTI DI BASI DI DATI E PROGRAMMAZIONE AD OGGETTI
ANNO ACCADEMICO 2022/2023

Progettazione e sviluppo di una base di dati relazionale per la descrizione e la memorizzazione di conferenze scientifiche

Autori:

ALESSANDRO GRIECO

Matricola N86/4241

alessandro.grieco2@studenti.unina.it

GIANFRANCO DUMINUOCO

Matricola N86/4061

gianfranco.duminuco@studenti.unina.it

Docenti:

Prof. Mara SANGIOVANNI

Indice

1	PREFAZIONE	2
1.1	Conferenza, sede, programma, locazione e sessione	2
1.2	Evento sociale e intervallo	2
1.3	Organizzatore locale, organizzatore scientifico, partecipante, intervento ed ente . . .	2
1.4	Sponsor e pubblicità	2
2	PROGETTAZIONE CONCETTUALE	3
2.1	ER	3
2.2	CLASS DIAGRAM	4
2.3	Ristrutturazione del class diagram	5
2.4	CLASS DIAGRAM RISTRUTTURATO	6
2.5	Dizionario classi	7
2.6	Dizionario associazioni	10
2.7	Dizionario vincoli	13
3	Progettazione Logica	15
3.1	Schema Logico	15
4	PROGETTAZIONE FISICA	17
4.1	Creazione delle enumerazioni	17
4.2	Creazione delle tabelle	17
4.2.1	Tabella SEDE	17
4.2.2	Tabella LOCAZIONE	17
4.2.3	Tabella CONFERENZA	18
4.2.4	Tabella SPONSOR	18
4.2.5	Tabella PROGRAMMA	18
4.2.6	Tabella INTERVALLO	19
4.2.7	Tabella EVENTO_SOCIALE	20
4.2.8	Tabella ENTE	21
4.2.9	Tabella ORGANIZZATORE_LOCALE	21
4.2.10	Tabella ORGANIZZATORE_SCIENTIFICO	21
4.2.11	Tabella PARTECIPANTE	22
4.2.12	Tabella SESSIONE	22
4.2.13	Tabella INTERVENTO	23
4.2.14	Tabella PUBBLICITA'	24
4.2.15	Tabella AMMINISTRARE	25
4.2.16	Tabella ORGANIZZARE_L	25
4.2.17	Tabella ORGANIZZARE_S	26
4.2.18	Tabella PARTECIPARE	26
4.3	Trigger,Procedure e View	27
4.3.1	AUTOMATISMO AMMINISTRARE ORGANIZZATORE	27
4.3.2	AUTOMATISMO PARTECIPAZIONE KEYNOTESPEAKER	28
4.3.3	TRIGGER CONFLISSO SESSIONI	29
4.3.4	TRIGGER LOCAZIONE VALIDA	30
4.4	TRIGGER CONFLITTO CONFERENZE	31
4.4.1	VIEW RIEPILOGOKS	32
4.5	Popolamento del database	33

1 PREFAZIONE

Si provvederà alla progettazione e sviluppo di una base di dati dedicata alla gestione di conferenze. La base di dati permetterà di effettuare modifiche a conferenze già esistenti, di inserire conferenze future e visualizzare specifici dati riguardanti tutto ciò di cui una conferenza è composta.

1.1 Conferenza, sede, programma, locazione e sessione

Questo insieme di entità sono lo scheletro di una conferenza, permettono di identificare le informazioni sulla data/orario/luogo della conferenza:

- **conferenza** : Fornisce dati generali sulla trattazione e organizzazione temporale della conferenza;
- **programma e sessione** : Consentono di conoscere gli orari e argomenti delle varie sedute della conferenza;
- **sede e locazione** : Definiscono il luogo fisico in cui si terranno le varie sessioni di una conferenza.

1.2 Evento sociale e intervallo

Consentono ad un partecipante e/o ad un organizzatore di identificare i momenti di pausa ed i momenti di eccezione di una conferenza

1.3 Organizzatore locale, organizzatore scientifico, partecipante, intervento ed ente

Questo gruppo di entità costituisce la base delle varie sessioni di una conferenza. Compongono la popolazione di una conferenza, come questa partecipa alle varie sessioni e come questa si muove all'interno di una conferenza.

- **ente** : Specifica l'ente di appartenenza di ciascun elemento appartenente alla popolazione di una conferenza;
- **Organizzatore locale e scientifico** : Definisce le persone fisiche che organizzano una conferenza;
- **partecipante e intervento** : Il partecipante è lo spettatore fisico di una sessione, il quale può intervenire o meno all'interno di essa.

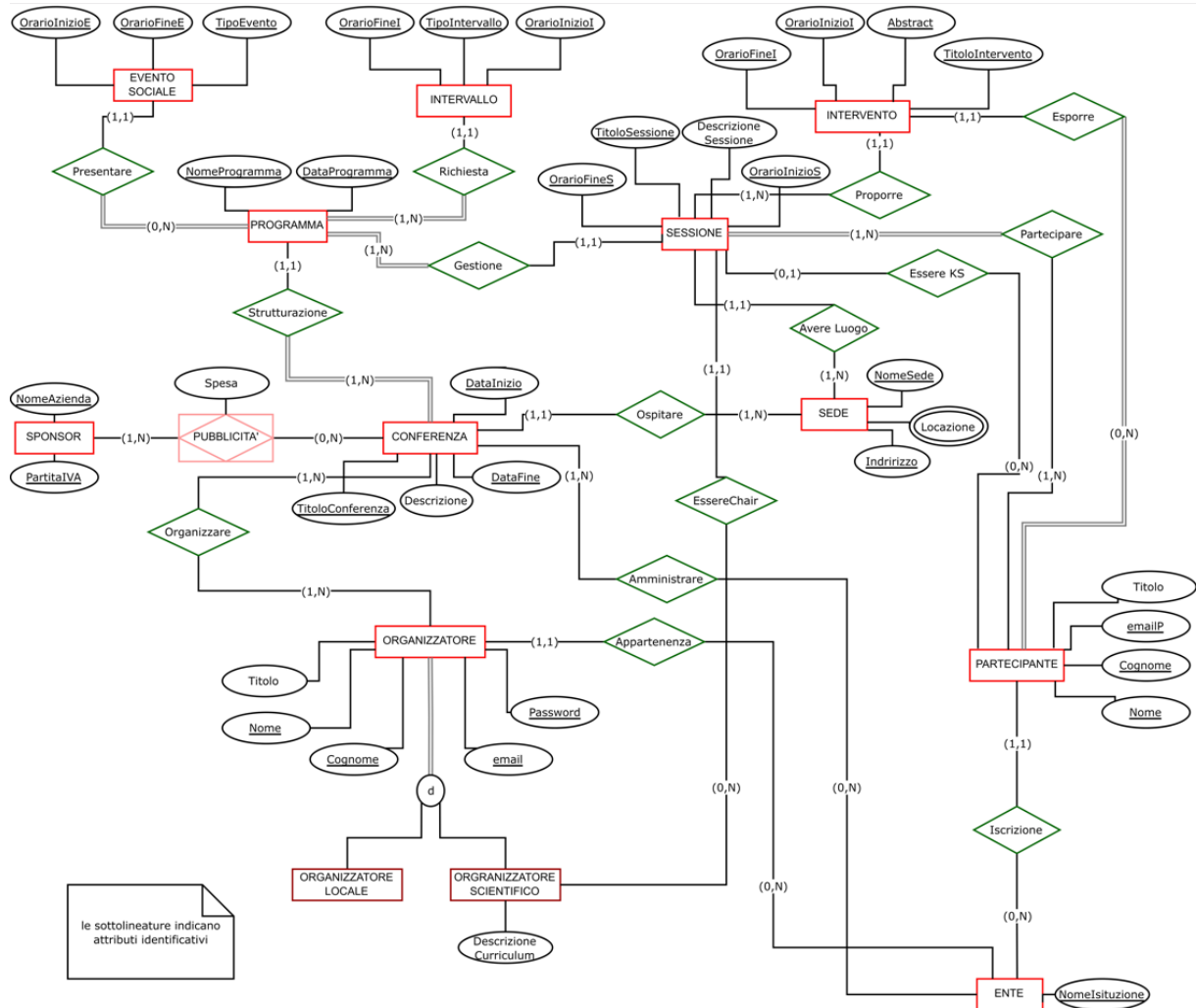
1.4 Sponsor e pubblicità

Determinano l'insieme delle aziende le quali contribuiscono (anche economicamente) alla realizzazione di una conferenza.

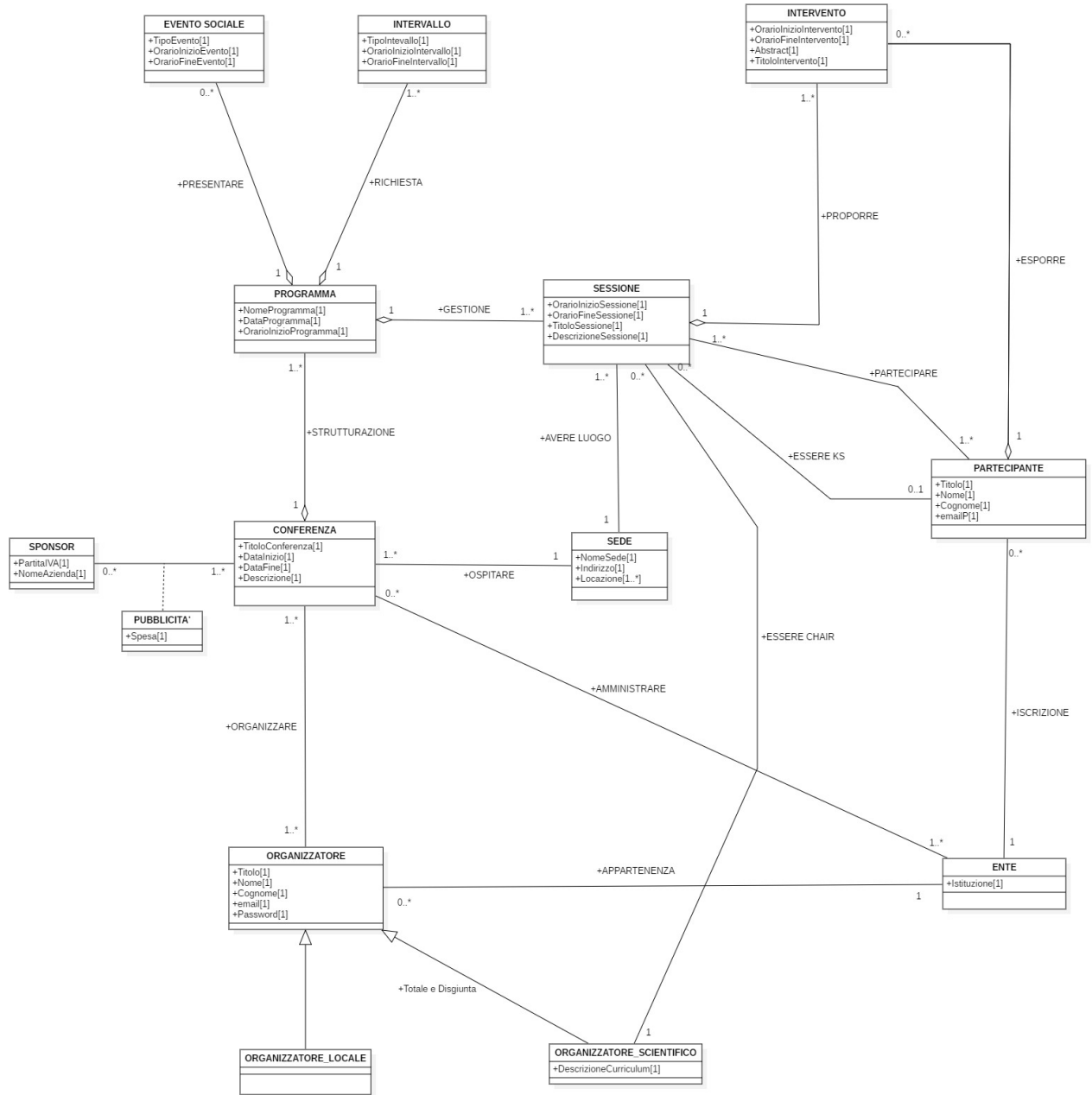
2 PROGETTAZIONE CONCETTUALE

In seguito sono riportati il diagramma ER e il rispettivo class diagram:

2.1 ER



2.2 CLASS DIAGRAM



2.3 Ristrutturazione del class diagram

Per motivi di implementazione fisica è necessario ristrutturare il class diagram sopracitato 2.2, tenendo conto dei seguenti punti:

Eliminazione attributi multivalore L'unico attributo multivalore presente nel diagramma risulta essere l'attributo locazione nell'entità sede, quindi è necessario rendere quest'ultimo un'entità a sé stante.

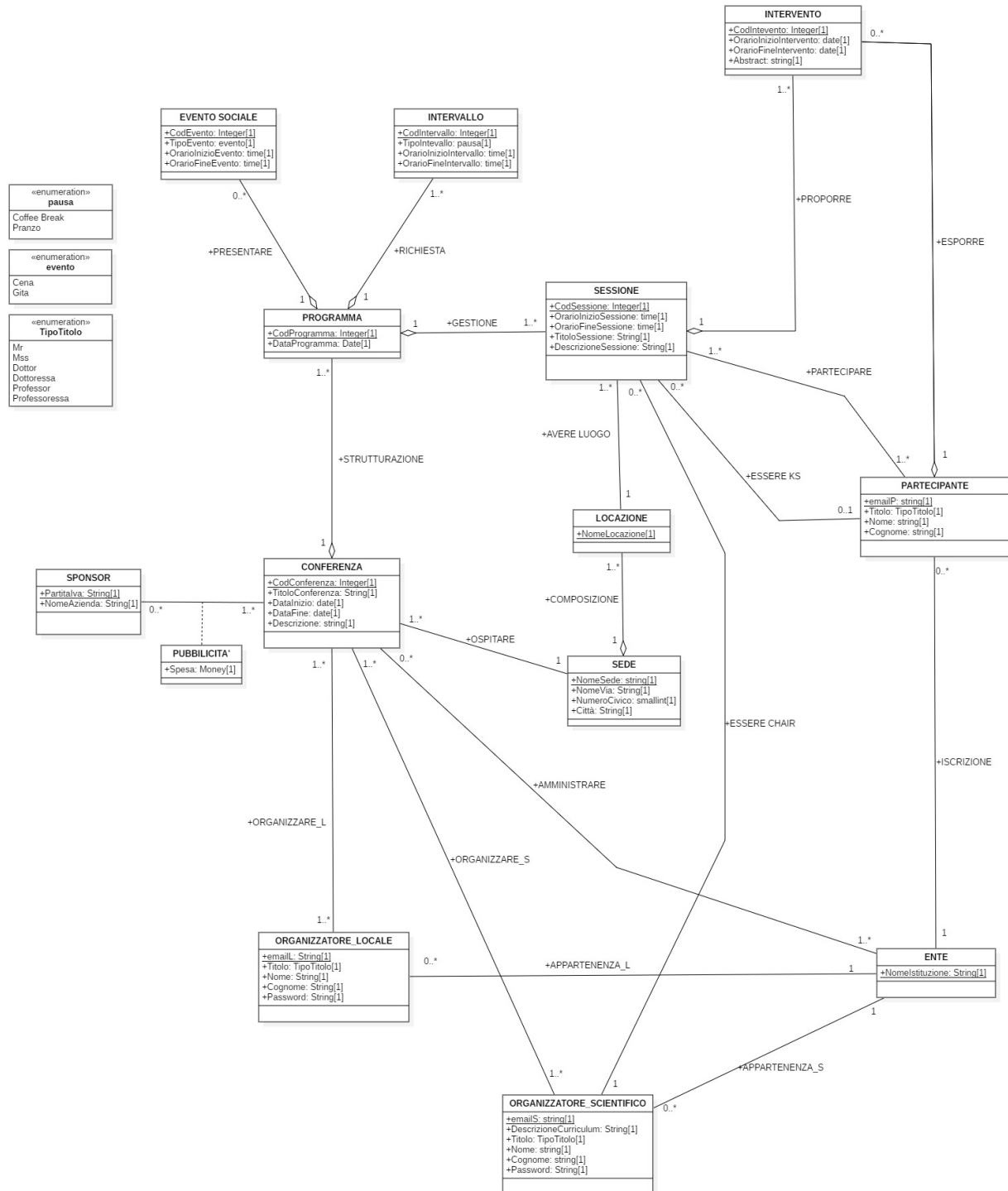
Eliminazione attributi strutturati L'unico attributo strutturato risulta essere l'attributo indirizzo all'interno dell'entità sede, il quale può essere ristrutturato componendolo nei suoi sotto-attributi (NomeVia, NumeroCivico, Città)

Scelta degli identificatori primari Per le entità sessione, programma, conferenza, intervento, intervallo, evento sociale sono state introdotte delle chiavi surrogate (rispettivamente codsessione, codprogramma, codconferenza, codintervento, codintervallo, codevento). Per le entità sponsor, partecipante, organizzatore locale, organizzatore scientifico, ente, locazione e sede sono stati identificate come chiavi attributi preesistenti (rispettivamente PartitaIva, emailP, emailL, emailS, NomeIstituzione, NomeLocazione, NomeSede).

Eliminazioni delle gerarchie L'unica gerarchia identificata (totale e disgiunta) nel class diagram è quella tra organizzatore, organizzatore scientifico ed organizzatore locale. Poiché l'entità organizzatore scientifico è totale e disgiunta, è possibile andare ad accorpare gli attributi dell'entità padre all'interno delle entità figlie.

Analisi delle ridondanze Nel class diagram non sono presenti ridondanze.

2.4 CLASS DIAGRAM RISTRUTTURATO



2.5 Dizionario classi

CLASSE	DESCRIZIONE	ATTRIBUTI
CONFERENZA	Riunione di rappresentanti di vari enti, per uno scambio di vedute su problemi di attualità o per risolvere una controversia internazionale	CodConferenza (integer): Chiave tecnica. Identifica univocamente ciascuna istanza di Conferenza. DataInizio (date) : Data in cui ha luogo la Conferenza. DataFine (date) : Data ove termina la Conferenza. Descrizione (string) : Testo che illustra gli obiettivi della Conferenza. TitoloConferenza (string) : Nome che identifica la Conferenza.
SEDE	Luogo fisico ove si tiene la conferenza	NomeSede (string) : Nome identificativo della sede. Identifica univocamente le istanze di Sede. NomeVia (string) : Nome nella strada in cui è presente la Sede. NumeroCivico (integer): Numero effettivo della strada Città (string): Comune dove è presente la Sede
LOCAZIONE	Struttura interna alla sede ove si tiene la sessione	NomeLocazione (string) : Nome della locazione
PROGRAMMA	Gestore della conferenza per migliorare la programmazione delle sedute della conferenza	CodProgramma (integer) : Chiave tecnica. Identifica univocamente ciascuna istanza di Programma. DataProgramma (date) : Data in cui è programmata una giornata della conferenza
SESSIONE	Frammento di una conferenza avvenuta in una data in un certo orario	CodSessione (integer) : Chiave tecnica. Identifica univocamente le istanze di Sessione. OrarioInizioSessione (time) : Orario inizio della Sessione OrarioFineSessione (time) : Orario fine della Sessione. TitoloSessione (string) : Denominazione identificatrice della Sessione DescrizioneSessione (string) : Testo illustrativo degli obiettivi e discussione della Sessione.

CLASSE	DESCRIZIONE	ATTRIBUTI
PARTECIPANTE	Persona che partecipa alla sessione in veste di partecipante	emailP (string): Indirizzo di posta elettronica del Partecipante. Identifica univocamente le istanze di partecipante. Titolo (TipoTitolo) : Carica accademica, professionale o scientifica di rilievo (ex: dott./dott.ssa). Nome (string) : Nome del partecipante. Cognome (string) : Cognome del partecipante.
INTERVENTO	L'atto di prendere la parola al fine di metter luce su argomenti inerenti alla sessione	CodIntervento (string) : Chiave tecnica. Identifica univocamente le istanze di Intervento. OrarioInizioIntervento (time) : Orario che ha inizio l'Intervento. OrarioFineIntervento (time) : Orario che termine l'Intervento. Abstract (string) : Breve testo il cui contenuto descrive cosa verrà presentato dall'Intervento.
PUBBLICITA'	Identifica il costo a spese dell'azienda nel farsi pubblicità nella conferenza.	Spesa (Money) : Costo pubblicitario per l'azienda
EVENTO SOCIALE	Celebrazione o commemorazione di un particolare momento trascorso tra persone interessate alla tematica della conferenza	CodEvento (integer) : Chiave tecnica. Identifica univocamente le istanze di Evento Sociale. TipoEvento (evento) : Enumerazione che specifica il tipo di evento (ex: Cena). OrarioInizioEvento (time) : Orario d'inizio dell'evento sociale. OrarioFineEvento (time) : Orario fine dell'evento sociale.
INTERVALLO	Breve interruzione di un programma o sessione che consente ai partecipanti di riposarsi prima della sua continuazione	CodIntervallo (integer) : Chiave tecnica. Identifica univocamente le istanze di Intervallo. TipoIntervallo (pausa) : Enumazione che specifica il tipo di intervallo (ex: coffee break) OrarioInizioIntervallo (time) : Orario d'inizio dell'Intervallo. OrarioFineIntervallo (time) : Orario di fine Intervallo

CLASSE	DESCRIZIONE	ATTRIBUTI
ORGANIZZATORE SCIENTIFICO	Responsabile della gestione su tematiche scientifiche e saccente su uno dei campi che afferiscono alla conferenza	<p>emailS (string) : Indirizzo di posta elettronica. Identifica univocamente le istanze di Organizzatore Scientifico.</p> <p>DescrizioneCurriculum (string) : Presentazione delle esperienze professionali e competenze di un Organizzatore Scientifico.</p> <p>Titolo (TipoTitolo) :Carica accademica o professionale o personale di rilievo (ex: dott./dt.ssa).</p> <p>Nome (string): Nome dell'organizzatore scientifico</p> <p>Cognome (string) : Cognome dell'organizzatore scientifico</p> <p>Password (string) : Parola necessaria per gestire l'accesso al programma applicativo</p>
ORGANIZZATORE LOCALE	Responsabile della gestione tecnica e logistica della conferenza	<p>emailL (string) : Indirizzo di posta elettronica. Identifica univocamente le istanze di organizzatore locale.</p> <p>Titolo (TipoTitolo) : Carica accademica, professionale o personale di rilievo.</p> <p>Nome (string) :Nome dell'organizzatore locale.</p> <p>Cognome (string) : Cognome dell'organizzatore locale.</p> <p>Password (string) : Parola necessaria per gestire l'accesso al programma applicativo</p>
ENTE	Istituzioni che ideano la conferenza definendone gli organizzatori	NomeIstituzione (string) : Nome dell'istituto che gestisce la conferenza.
SPONSOR	Azienda interessata a promuoversi pubblicizzandosi attraverso un sostegno finanziario	<p>PartitaIva (string) : Codice identificativo di chi svolge un'attività d'impresa o di lavoro autonomo.</p> <p>NomeAzienda (string) : Nome dell'azienda.</p>

2.6 Dizionario associazioni

NOME	DESCRIZIONE	CLASSI COINVOLTE
Strutturazione	Definisce la struttura di una conferenza in base al programma assegnatogli.	Conferenza[1..*] (<i>è strutturata da</i>): specifica i programmi di una Conferenza. Programma[1] (<i>struttura</i>): specifica a quale Conferenza è assegnato il nostro Programma.
Amministrare	Definisce la relazione tra le conferenze e gli enti organizzatori delle stesse.	Conferenza[1..*] (<i>è amministrata da</i>): specifica quali enti contribuiscono nell'organizzazione della conferenza. Ente[0..*] (<i>amministra</i>): specifica quali conferenze sono amministrate da un ente.
Essere Chair	Definisce il ruolo "essere chair" da parte di un organizzatore scientifico in realzione con una sessione.	Sessione[1] (<i>ha come chair</i>): specifica chi è il chair di una determinata sessione. Organizzatore_Scientifico[0..*] (<i>è chair di</i>): specifica di quali sessioni un organizzatore è chair.
Pubblicizza	Definisce la relazione di sponsorizzazione tra le conferenze e i suoi sponsor.	Sponsor[1..*] (<i>pubblicizza</i>):specifica quali conferenze sono pubblicizzate da uno sponsor. Conferenza[0..*] (<i>è pubblicizzata da</i>): specifica da quale sponsor una conferenza è pubblicizzata.
Organizzare_L	Definisce la relazione tra le conferenze e gli organizzatori locali che le organizzano.	Organizzatore_Locale[1..*] (<i>è organizzatore locale di</i>): specifica quali sono le conferenze che un organizzatore locale gestisce. Conferenza[1..*] (<i>ha come organizzatore locale</i>): specifica chi sono gli organizzatori locali che gestiscono la conferenza.
Organizzare_S	Definisce la relazione tra le conferenze e gli organizzatori scientifici che la organizzano.	Organizzatore_Scientifico[1..*] (<i>è organizzatore scientifico di</i>): specifica quali sono le conferenze che un organizzatore locale gestisce. Conferenza[1..*] (<i>ha come organizzatore scientifico</i>): specifica chi sono gli organizzatori scientifici che gestiscono la conferenza.
Appartenenza_L	Definisce la relazione di appartenenza tra gli organizzatore locali e le loro istituzioni.	Organizzatore_Locale[1] (<i>appartiene a</i>): specifica a quale ente appartiene un organizzatore locale. Ente[0..*] (<i>ha come organizzatore locale</i>): specifica chi sono gli organizzatori locali che appartengono all'ente.
Appartenenza_S	Definisce la relazione di appartenenza tra gli organizzatori scientifici e le loro istituzioni.	Organizzatore_Scieifico[1] (<i>appartiene a</i>): specifica a quale ente appartiene un organizzatore scientifico. Ente[0..*] (<i>ha come organizzatore scientifico</i>): specifica chi sono gli organizzatori scientifici che appartengono all'ente.

NOME	DESCRIZIONE	CLASSI COINVOLTE
Iscrizione	Definisce il ruolo "essere iscritto" tra i partecipanti e gli istituti a cui sono iscritti	Partecipante[1] (<i>è iscritto a</i>): specifica a quale ente appartiene un partecipante di una sessione. Ente[0..*] (<i>ha come iscritto</i>): specifica chi sono i partecipanti che sono iscritti all'ente.
Gestione	Definisce la relazione tra un programma di una conferenza e le sessioni che questo gestisce	Sessione[1] (<i>è gestita da</i>): specifica da quale programma è gestita la sessione. Programma[1..*] (<i>gestisce</i>): specifica quali sessioni gestisce il programma della conferenza.
Presentare	Definisce la relazione tra un programma di una conferenza e gli eventi sociali che questo presenta	EventoSociale[1] (<i>è presentato da</i>): specifica da quale programma è presentato l'evento sociale. Programma[0..*] (<i>presenta</i>): specifica quali Eventi sociali presenta il programma della conferenza.
Richiesta	Definisce la relazione tra un programma di una conferenza e gli intervalli che questo richiede	Intervallo[1] (<i>è richiesto in</i>): specifica da quale programma è richiesto l'intervallo. Programma[1..*] (<i>richiede</i>): specifica quali intervalli il programma richiede.
Proporre	Definisce la relazione tra una sessione e gli interventi che questa propone	Intervento[1] (<i>è proposto in</i>): specifica in quale sessione è proposto l'intervento. Sessione[1..*] (<i>propone</i>): specifica quali interventi una sessione propone.
Essere KS	Definisce la relazione "essere keynote speaker" tra un partecipante che è KS e le sessioni in cui lo è.	Sessione[0..1] (<i>ha come KS</i>): specifica chi è il KeynoteSpeaker di una sessione(se esiste). Partecipante[0..*] (<i>è KS di</i>): specifica di quale sessione il partecipante è KS.
Ospitare	Definisce la relazione tra una sede e le conferenze che vengono ospitate.	Conferenza[1] (<i>è ospitata</i>): specifica in quale sede è ospitata la conferenza. Sede[1..*] (<i>ospita</i>): specifica quali conferenze la sede ospita.
Composizione	Definisce la relazione tra una sede e le locazioni che la compongono.	Locazione[1] (<i>compone</i>): specifica quale sede compone la locazione. Sede[1..*] (<i>è composta</i>): specifica con quali locazioni è composta una sede.
Avere luogo	Definisce la relazione tra una sessione e le locazioni in cui ha luogo.	Sessione[1] (<i>ha luogo in</i>): specifica in quale locazione ha luogo la sede. Locazione[1..*] (<i>prende in luogo</i>): specifica quali sessioni prendono luogo in una locazione.
Partecipare	Definisce la relazione tra le sessioni e i suoi partecipanti.	Sessione[1..*] (<i>ha come partecipanti</i>): specifica chi sono i partecipanti di una sessione. Partecipante[1..*] (<i>partecipa a</i>): specifica a quali sessioni un partecipante partecipa.

NOME	DESCRIZIONE	CLASSI COINVOLTE
Esporre	Definisce la relazione tra un intervento e il partecipante che lo espone.	Intervento[1] (<i>è esposto da</i>): specifica quale partecipante espone un intervento. Partecipante[0..*] (<i>espone</i>): specifica quali sono gli interventi che un partecipante espone.

2.7 Dizionario vincoli

NOME VINCOLO	TIPO	DESCRIZIONE
EmailValida	Dominio	Le email devono contenere una "@", minimo due caratteri prima della @, ,minimo tre caratteri tra essa ed il punto, ed almeno DUE caratteri dopo il punto (ex: example@dominio.it)
NoConferenzaParallela	Intrarelazione	Non possono esserci più conferenze con le stesse date nella stessa sede.
ConflittoSessioni	Intrarelazionale	In una conferenza, non possono esserci più sessioni con gli stessi orari e con la stessa locazione programmate nella stessa data
NomeAziendaUnico	Intrarelazionale	Il nome dell'azienda deve essere univoco
KSPartecipante	Interrelazionale	Il keynote speaker di una sessione deve essere necessariamente partecipante di quella sessione
InterventoInOrario	Interrelazionale	L'orario d'inizio e l'orario di fine intervento devono essere compresi tra l'orario d'inizio e l'orario di fine della sessione a cui l'intervento è legato.
DataProgrammaValida	Interrazionale	La data di un programma deve essere compresa (o uguale) tra la data inizio e data fine della conferenza che sta strutturando.
DataProgrammaUnica	Intrarelazionale	Può esserci un solo programma per un giorno della conferenza
DataConferenzaValida	N-upla	La data di inizio conferenza deve essere minore della data di fine conferenza
OrarioIntervalloValido	N-upla	L'orario di inizio di un intervallo deve essere minore dell'orario di fine intervallo
OrarioSessioneValido	N-upla	L'orario di inizio sessione deve essere minore dell'orario di fine sessione
OrarioInterventoValido	N-upla	L'orario di inizio intervento deve essere minore dell'orario di fine intervento.
OrarioEventoSocialeValido	N-upla	L'orario di inizio evento sociale deve essere minore dell'orario di fine evento sociale.
IndirizzoUnico	Intrarelazionale	Ogni sede ha un unico indirizzo
ConflittoSessioneEventoSociale	Interrelazionale	Una sessione non può iniziare o finire durante un evento sociale che avviene nella sua stessa data.
ConflittoEventoSocialeSessione	Interrelazionale	Un evento sociale non può iniziare o finire mentre è in corso una sessione che avviene nella stessa data

NOME VINCOLO	TIPO	DESCRIZIONE
ConflittoIntervalloSessione	Interrelazionale	Un intervallo non può iniziare o finire mentre è in corso una sessione che avviene nella sua stessa data
ConflittoSessioneIntervallo	Interrelazionale	Una sessione non può iniziare o finire mentre è in corso un intervallo che avviene nella sua stessa data
ConflittoIntervalloEventoSociale	Interrelazionale	Un intervallo non può iniziare o finire mentre è in corso un evento sociale che avviene nella sua stessa data
ConflittoEventoSocialeIntervallo	Interrelazionale	Un evento sociale non può iniziare o finire mentre è in corso un intervallo che avviene nella sua stessa data.
organizzareEnte_I	Interrelazionale	L'organizzatore di una conferenza deve necessariamente appartenere a un ente che amministra la conferenza.
organizzareEnte_S	Interrelazionale	L'organizzatore di una conferenza deve necessariamente appartenere a un ente che amministra la conferenza.
LocazioneValida	Interrelazionale	La locazione di una sessione deve appartenere alla sede che ospita la conferenza di quella sessione.
ConflittoEventoSociale	Intrarelazionale	In una conferenza, un evento sociale non può iniziare mentre è in corso un altro evento sociale che avviene nello stessa data.
ConflittoIntervalli	Intrarelazionale	Un intervallo non può iniziare o finire mentre è in corso un altro intervallo nello stessa data.
ConflittoInterventi	Intrarelazionale	In una sessione, un intervento non può iniziare o finire mentre è in corso un altro intervento.
ChairValido	Interrelazionale	Il char deve essere un organizzatore scientifico della conferenza nella quale è presente la sessione che sta amministrando.

3 Progettazione Logica

In seguito verrà riportate la traduzione dallo schema concettuale allo schema logico. Si noti che gli attributi con singola sottolineatura rappresentano la primary key della classe considerata, mentre gli attributi con doppia sottolineatura rappresentano le foreign key; dopo aver descritto la classe vengono quindi delineate le entità a cui le chiavi esterne si riferiscono.

3.1 Schema Logico

CLASSE	ATTRIBUTI
CONFERENZA	(<u>CodConferenza</u> , DataInizio, DataFine, Descrizione, TitoloConferenza, <u>NomeSede</u>)
	<i>Chiave esterna:</i> NomeSede → Sede.NomeSede
PROGRAMMA	(<u>CodProgramma</u> , DataProgramma, <u>CodConferenza</u> ,)
	<i>Chiave esterna:</i> CodConferenza → Conferenza.CodConferenza
SESSIONE	(<u>CodSessione</u> , OrarioInizioSessione, OrarioFineSessione, TitoloSessione, DescrizioneSessione, <u>Chair</u> , <u>KeynoteSpeaker</u> , <u>CodProgramma</u> , <u>NomeLocazione</u>)
	KeynoteSpeaker → Partecipante.emailP
	<i>Chiave esterna:</i> CodProgramma → Programma.CodProgramma
	Chair → Organizzatore_Scientifico.emailS
	NomeLocazione → Locazione.NomeLocazione
PARTECIPANTE	(emailP, Titolo, Nome, Cognome, <u>Istituzione_di_Afferenza</u> , <u>CodIntervento</u>)
	<i>Chiave esterna:</i> CodIntervento → Intervento.CodIntervento
	Istituzione_di_Afferenza → Ente.NomeIstituzione
LOCAZIONE	(<u>NomeLocazione</u> , <u>NomeSede</u>)
	<i>Chiave esterna:</i> NomeSede → Sede.NomeSede
SEDE	(<u>NomeSede</u> , NomeVia, NumeroCivico, Città)
INTERVENTO	(<u>CodIntervento</u> , OrarioInizioIntervento, OrarioFineIntervento, Abstract, <u>CodPartecipante</u> , <u>CodSessione</u>)
	<i>Chiave esterna:</i> CodPartecipante → Partecipante.emailP
	CodSessione → Sessione.CodSessione

CLASSE	ATTRIBUTI
EVENTO SOCIALE	(<u>CodEvento</u> , TipoEvento, OrarioInizioEvento, OrarioFineEvento, <u>CodProgramma</u>)
	<i>Chiave esterna:</i> CodProgramma → Programma.CodProgramma
INTERVALLO	(<u>CodIntervallo</u> , TipoIntervallo, OrarioInizioIntervallo, OrarioFineIntervallo, <u>CodProgramma</u>)
	<i>Chiave esterna:</i> CodProgramma → Programma.CodProgramma
ORGANIZZATORE_ SCIENTIFICO	(<u>emailS</u> , DescrizioneCurriculum, Titolo, Nome, Cognome, <u>Istituzione di Afferenza</u>)
	<i>Chiave esterna:</i> Istituzione di Afferenza → Ente.NomeIstituzione
ORGANIZZATORE_ LOCALE	(<u>emailL</u> , Titolo, Nome, Cognome, <u>Istituzione di Afferenza</u>)
	<i>Chiave esterna:</i> Istituzione di Afferenza → Ente.Istituzione di Afferenza
ENTE	(<u>NomeIstituzione</u>)
SPONSOR	(<u>PartitaIva</u> , Spesa, NomeAzienda)
PUBBLICITA'	(<u>PartitaIva</u> , <u>CodConferenza</u> , Spesa)
	PartitaIva → Sponsor.PartitaIva CodConferenza → Conferenza.CodConferenza
ORGANIZZARE_ S	(<u>emailS</u> , <u>CodConferenza</u>)
	emailS → Organizzatore_ Scientifico.emailS CodConferenza → Conferenza.CodConferenza
ORGANIZZARE_ L	(<u>emailL</u> , <u>CodConferenza</u>)
	emailL → Organizzatore_ Locale.emailL CodConferenza → Conferenza.CodConferenza
AMMINISTRARE	(<u>NomeIstituzione</u> , <u>CodConferenza</u>)
	NomeIstituzione → Ente.NomeIstituzione CodConferenza → Conferenza.CodConferenza
PARTECIPARE	(<u>emailP</u> , <u>CodSessione</u>)

CLASSE	ATTRIBUTI
<i>Chiave esterna:</i>	emailP → Partecipante.emailP
	CodSessione → Sessione.CodSessione

4 PROGETTAZIONE FISICA

Per la realizzazione della progettazione fisica è stato scelto il DBMS **POSTGRESQL**, vi è quindi la possibilità di implementare il progetto sfruttando le capacità del DBMS (che queste differiscano o meno da analoghi softwares).

4.1 Creazione delle enumerazioni

```
CREATE TYPE TipoIntervallo AS ENUM ('CoffeBreak','Pranzo');

CREATE TYPE TipoEvento AS ENUM ('Cena','Gita');

CREATE TYPE TipoTitolo AS ENUM
('Mr','Mss','Dottor','Dottorressa','Professor','Professoressa');
```

4.2 Creazione delle tabelle

4.2.1 Tabella SEDE

```
--Creo la tabella SEDE:
CREATE TABLE SEDE
(
NomeSede VARCHAR(100) PRIMARY KEY,
NomeVia VARCHAR(100) NOT NULL,
NumeroCivico SMALLINT NOT NULL,
Citta VARCHAR(100) NOT NULL
);
```

4.2.2 Tabella LOCAZIONE

```
--Creo la tabella LOCAZIONE:
CREATE TABLE LOCAZIONE
(
NomeLocazione VARCHAR(100) PRIMARY KEY,
NomeSede VARCHAR(100) NOT NULL
);

--Aggiungo i vincoli:
--chiave esterna:
--Se una sede viene cancellata, vengono cancellate anche
--tutte le sue locazioni
ALTER TABLE LOCAZIONE ADD
CONSTRAINT Composizione FOREIGN KEY(NomeSede)
REFERENCES SEDE(NomeSede) ON DELETE CASCADE;
```

4.2.3 Tabella CONFERENZA

```
--Creo la tabella CONFERENZA:
CREATE TABLE CONFERENZA
(
  CodConferenza SERIAL PRIMARY KEY,
  DataInizio DATE NOT NULL,
  DataFine DATE NOT NULL,
  TitoloConferenza VARCHAR(64) NOT NULL,
  Descrizione VARCHAR(1500) NOT NULL,
  NomeSede VARCHAR(100) NOT NULL
);

--per l'attributo chiave CodConferenza e' stato applicato il tipo SERIAL
--affinche' si autogenerasse automaticamente una chiave primaria, tuttavia
--per le prossime tabelle nelle quali e' stato necessario introdurre
--chiavi autogenerate, per fini scolastici verranno introdotte delle automazioni
--create a mano implementate con dei TRIGGERS.

--chiave esterna:
ALTER TABLE CONFERENZA ADD
CONSTRAINT Ospitare FOREIGN KEY(NomeSede)
REFERENCES SEDE(NomeSede) ON DELETE CASCADE;

--imposto l'inizio della sequenza di generazione di codici conferenze a 1
ALTER SEQUENCE conferenza_codconferenza_seq RESTART WITH 1;
```

4.2.4 Tabella SPONSOR

```
--Creo la tabella SPONSOR:
CREATE TABLE SPONSOR
(
  PartitaIVA CHAR(11) PRIMARY KEY,
  NomeAzienda VARCHAR(50) NOT NULL
);
```

4.2.5 Tabella PROGRAMMA

```
--Creo la tabella PROGRAMMA:
CREATE TABLE PROGRAMMA
(
  CodProgramma INTEGER DEFAULT 0,
  DataProgramma DATE NOT NULL,
  CodConferenza INTEGER NOT NULL
);

--Aggiungo i vincoli:
--chiave primaria:
ALTER TABLE PROGRAMMA ADD
CONSTRAINT Programma_pk PRIMARY KEY(CodProgramma);
--chiave esterna:
--se una conferenza viene cancellata, vengono cancellati
--tutti i programmi legati a essa
ALTER TABLE PROGRAMMA ADD
CONSTRAINT Strutturazione FOREIGN KEY(CodConferenza)
REFERENCES CONFERENZA(CodConferenza) ON DELETE CASCADE;

--PK AUTOMATICA
--Definisco la funzione richiamata dal Trigger:
```

```

CREATE OR REPLACE FUNCTION GenerateProgrammaPK() RETURNS TRIGGER AS
$check$
DECLARE
    pk PROGRAMMA.CodProgramma%TYPE;
    Conta INTEGER;
BEGIN
    SELECT COUNT(CodProgramma) FROM PROGRAMMA    --nel caso in cui la tabella sia vuota
    INTO Conta                                     --la pk e' 0
    LIMIT 2;

    IF (Conta=0) THEN
        NEW.CodProgramma = 0;
        RETURN NEW;
    END IF;

    SELECT MAX(CodProgramma) FROM PROGRAMMA
    INTO pk;
    NEW.CodProgramma = pk + 1;
    RETURN NEW;
END;
$check$ LANGUAGE PLPGSQL;

--Definisco il Trigger per settare automaticamente la pk di EVENTO_SOCIALE:
CREATE OR REPLACE TRIGGER ProgrammaPK
BEFORE INSERT ON PROGRAMMA
FOR EACH ROW
EXECUTE PROCEDURE GenerateProgrammaPK();

```

4.2.6 Tabella INTERVALLO

```

--Creo la tabella INTERVALLO:
CREATE TABLE INTERVALLO
(
    CodIntervallo INTEGER DEFAULT 0,
    TipoIntervallo TipoIntervallo NOT NULL,
    OrarioInizioIntervallo TIME NOT NULL,
    OrarioFineIntervallo TIME NOT NULL,
    CodProgramma INTEGER NOT NULL
);

--Aggiungo vincoli:
--chiave primaria:
ALTER TABLE INTERVALLO ADD
CONSTRAINT Intervallo_pk PRIMARY KEY(CodIntervallo);
--chiave esterna:
--CodProgramma
--se viene cancellato un programma, vengono cancellate
--tutte le pause che erano richieste nel programma
ALTER TABLE INTERVALLO ADD
CONSTRAINT Richiesta FOREIGN KEY(CodProgramma)
REFERENCES PROGRAMMA(CodProgramma) ON DELETE CASCADE;

--PK AUTOMATICA
--Definisco la funzione richiamata dal Trigger:
CREATE OR REPLACE FUNCTION GenerateIntervalloPK() RETURNS TRIGGER AS
$check$
DECLARE
    pk INTERVALLO.CodIntervallo%TYPE;
    Conta INTEGER;
BEGIN

```

```

SELECT COUNT(CodIntervallo) FROM INTERVALLO      --nel caso in cui la tabella sia vuota
INTO Conta                                         --la pk e' 0
LIMIT 2;

IF(Conta=0) THEN
    NEW.CodIntervallo = 0;
RETURN NEW;
END IF;

SELECT MAX(CodIntervallo) FROM INTERVALLO
INTO pk;
NEW.CodIntervallo = pk + 1;

RETURN NEW;
END;
$check$ LANGUAGE PLPGSQL;

--Definisco il Trigger per settare automaticamente la pk di INTERVALLO:
CREATE OR REPLACE TRIGGER IntervalloPK
BEFORE INSERT ON INTERVALLO
FOR EACH ROW
EXECUTE PROCEDURE GenerateIntervalloPK();

```

4.2.7 Tabella EVENTO_SOCIALE

```

--Creo la tabella EVENTO_SOCIALE:
CREATE TABLE EVENTO_SOCIALE
(
    CodEvento INTEGER DEFAULT 0,
    TipoEvento TipoEvento NOT NULL,
    OrarioInizioEvento TIME NOT NULL,
    OrarioFineEvento TIME NOT NULL,
    CodProgramma INTEGER NOT NULL
);

--Aggiungo i vincoli:
--chiave primaria:
ALTER TABLE EVENTO_SOCIALE ADD
CONSTRAINT Evento_Sociale_pk PRIMARY KEY(CodEvento);
--chiave esterna:
--CodProgramma
--se viene cancellato un programma, vengono cancellati
--tutti gli eventi sociali presentati dal programma
ALTER TABLE EVENTO_SOCIALE ADD
CONSTRAINT Presentare FOREIGN KEY(CodProgramma)
REFERENCES PROGRAMMA(CodProgramma) ON DELETE CASCADE;

--PK AUTOMATICA
--Definisco la funzione richiamata dal Trigger:
CREATE OR REPLACE FUNCTION GenerateEvento_SocialePK() RETURNS TRIGGER AS
$check$
DECLARE
    pk EVENTO_SOCIALE.CodEvento%TYPE;
    Conta INTEGER;
BEGIN
    SELECT COUNT(CodEvento) FROM EVENTO_SOCIALE      --nel caso in cui la tabella sia vuota
    INTO Conta                                         --la pk e' 0
    LIMIT 2;

```

```

    IF(Conta=0) THEN
        NEW.CodEvento = 0;
    RETURN NEW;
    END IF;

    SELECT MAX(CodEvento) FROM EVENTO_SOCIALE
    INTO pk;
    NEW.CodEvento = pk + 1;
    RETURN NEW;
END;
$check$ LANGUAGE PLPGSQL;

--Definisco il Trigger per settare automaticamente la pk di EVENTO_SOCIALE:
CREATE OR REPLACE TRIGGER Evento_SocialePK
BEFORE INSERT ON EVENTO_SOCIALE
FOR EACH ROW
EXECUTE PROCEDURE GenerateEvento_SocialePK();

```

4.2.8 Tabella ENTE

```

--Creo la tabella ENTE
CREATE TABLE ENTE
(
    NomeIstituzione VARCHAR(100) PRIMARY KEY
);

```

4.2.9 Tabella ORGANIZZATORE_LOCALE

```

--Creo la tabella Organizzatore_Locale:
CREATE TABLE Organizzatore_Locale
(
    email VARCHAR(100) PRIMARY KEY CHECK(email LIKE '___@___%.___%'),
    password VARCHAR(25) NOT NULL,
    Titolo TipoTitolo NOT NULL,
    Nome VARCHAR(100) NOT NULL,
    Cognome VARCHAR(100) NOT NULL,
    Istituzione_Di_Afferenza VARCHAR(100) NOT NULL
);

--Aggiungo vincoli:
--chiavi esterne:
--Istituzione_Di_Afferenza
ALTER TABLE Organizzatore_Locale ADD
CONSTRAINT Appartenenza_L FOREIGN KEY(Istituzione_Di_Afferenza)
REFERENCES ENTE(NomeIstituzione);

```

4.2.10 Tabella ORGANIZZATORE_SCIENTIFICO

```

--Creo la tabella Organizzatore_Scientifico:
CREATE TABLE Organizzatore_Scientifico
(
    emails VARCHAR(100) PRIMARY KEY CHECK(emails LIKE '___@___%.___%'),
    password VARCHAR(25) NOT NULL,
    DescrizioneCurriculum VARCHAR(500),
    Titolo TipoTitolo NOT NULL,
    Nome VARCHAR(100) NOT NULL,
    Cognome VARCHAR(100) NOT NULL,
    Istituzione_Di_Afferenza VARCHAR(100) );

```

```

--Aggiungo vincoli:
--chiavi esterne:
--Istituzione_Di_Afferenza
ALTER TABLE Organizzatore_Scientifico ADD
CONSTRAINT Appartenenza_S FOREIGN KEY(Istituzione_Di_Afferenza)
REFERENCES ENTE(NomeIstituzione);

```

4.2.11 Tabella PARTECIPANTE

```

--Creo la tabella PARTECIPANTE:
CREATE TABLE PARTECIPANTE
(
emailP VARCHAR(100) PRIMARY KEY CHECK(emailP LIKE '__%@__%.__%'),
Titolo TipoTitolo NOT NULL,
Nome VARCHAR(100) NOT NULL,
Cognome VARCHAR(100) NOT NULL,
Istituzione_di_Afferenza VARCHAR(100) NOT NULL
);

--Aggiungo i vincoli:
--chiavi esterne:
--Istituzione_di_Afferenza
ALTER TABLE PARTECIPANTE ADD
CONSTRAINT Iscrizione FOREIGN KEY(Istituzione_di_Afferenza)
REFERENCES ENTE(NomeIstituzione);

```

4.2.12 Tabella SESSIONE

```

--Creo la tabella SESSIONE:
CREATE TABLE SESSIONE
(
CodSessione INTEGER DEFAULT 0,
OrarioInizioSessione TIME NOT NULL,
OrarioFineSessione TIME NOT NULL,
TitoloSessione VARCHAR(100) NOT NULL,
DescrizioneSessione VARCHAR(10000),
Chair VARCHAR(100) NOT NULL,
KeyNoteSpeaker VARCHAR(100),
CodProgramma INTEGER NOT NULL,
NomeLocazione VARCHAR(100) NOT NULL
);

--Aggiungo i vincoli:
--chiave primaria:
ALTER TABLE SESSIONE ADD
CONSTRAINT CodSessione_pk PRIMARY KEY(CodSessione);
--chiavi esterne:
--Chair
ALTER TABLE SESSIONE ADD
CONSTRAINT EssereChair FOREIGN KEY(Chair)
REFERENCES ORGANIZZATORE_SCIENTIFICO(emails);
--KeyNoteSpeaker
--se viene cancellato il partecipante che era KS
--viene settato KS a NULL(la sessione non ha KS)
ALTER TABLE SESSIONE ADD
CONSTRAINT EssereKS FOREIGN KEY(KeyNoteSpeaker)
REFERENCES PARTECIPANTE(emailP) ON DELETE SET NULL;
--CodProgramma

```

```

--se viene cancellato un programma, vengono
--cancellate anche le sessioni legate al programma
ALTER TABLE SESSIONE ADD
CONSTRAINT Gestione FOREIGN KEY(CodProgramma)
REFERENCES PROGRAMMA(CodProgramma) ON DELETE CASCADE;
--NomeLocazione
ALTER TABLE SESSIONE ADD
CONSTRAINT AvereLuogo FOREIGN KEY(NomeLocazione)
REFERENCES LOCAZIONE(NomeLocazione);

--PK AUTOMATICA
--Definisco la funzione richiamata dal Trigger:
CREATE OR REPLACE FUNCTION GenerateSessionePK() RETURNS TRIGGER AS
$check$
DECLARE
    pk SESSIONE.CodSessione%TYPE;
    Conta INTEGER;
BEGIN
    SELECT COUNT(CodSessione) FROM SESSIONE      --nel caso in cui la tabella sia vuota
    INTO Conta                                     --la pk e' 0
    LIMIT 2;                                       --limito 2 tuple per risparmiare efficienza di calcolo

    IF(Conta=0) THEN
        NEW.CodSessione = 0;
        RETURN NEW;
    END IF;

    SELECT MAX(CodSessione) FROM SESSIONE
    INTO pk;
    NEW.CodSessione = pk + 1;
    RETURN NEW;
END;
$check$ LANGUAGE PLPGSQL;

--Definisco il Trigger per settare automaticamente la pk di CONFERENZA:
CREATE OR REPLACE TRIGGER SessionePK
BEFORE INSERT ON SESSIONE
FOR EACH ROW
EXECUTE PROCEDURE GenerateSessionePK();

```

4.2.13 Tabella INTERVENTO

```

--Creo la tabella INTERVENTO:
CREATE TABLE INTERVENTO
(
    CodIntervento INTEGER DEFAULT 0,
    OrarioInizioIntervento TIME NOT NULL,
    OrarioFineIntervento TIME NOT NULL,
    Abstract VARCHAR(500),
    CodPartecipante VARCHAR(100) NOT NULL,
    CodSessione INTEGER NOT NULL
);

--Aggiungo i vincoli:
--chiave primaria:
ALTER TABLE INTERVENTO ADD
CONSTRAINT Intervento_pk PRIMARY KEY(CodIntervento);
--chiavi esterne:
--CodPartecipante
--se un partecipante viene cancellato,

```



```

--vengono cancellati anche gli interventi
--legati a quel partecipante
ALTER TABLE INTERVENTO ADD
CONSTRAINT Esporre FOREIGN KEY(CodPartecipante)
REFERENCES PARTECIPANTE(emailP) ON DELETE CASCADE;
--CodSessione
--se una sessione viene cancellata, vengono
--cancellate tutti gli interventi proposti da
--quella sessione
ALTER TABLE INTERVENTO ADD
CONSTRAINT Proporre FOREIGN KEY(CodSessione)
REFERENCES SESSIONE(CodSessione) ON DELETE CASCADE;

--PK AUTOMATICA
--Definisco la funzione richiamata dal Trigger:
CREATE OR REPLACE FUNCTION GenerateInterventoPK() RETURNS TRIGGER AS
$check$
DECLARE
    pk INTERVENTO.CodIntervento%TYPE;
    Conta INTEGER;
BEGIN
    SELECT COUNT(CodIntervento) FROM INTERVENTO      --nel caso in cui la tabella sia vuota
    INTO Conta                                       --la pk e' 0
    LIMIT 2;

    IF (Conta=0) THEN
        NEW.CodIntervento = 0;
        RETURN NEW;
    END IF;

    SELECT MAX(CodIntervento) FROM INTERVENTO
    INTO pk;
    NEW.CodIntervento = pk + 1;
    RETURN NEW;
END;
$check$ LANGUAGE PLPGSQL;

--Definisco il Trigger per settare automaticamente la pk di INTERVENTO:
CREATE OR REPLACE TRIGGER InterventoPK
BEFORE INSERT ON INTERVENTO
FOR EACH ROW
EXECUTE PROCEDURE GenerateInterventoPK();

```

4.2.14 Tabella PUBBLICITA'

```

--Creo la tabella PUBBLICITA'
CREATE TABLE Pubblicita
(
    PartitaIva CHAR(11) NOT NULL,
    CodConferenza INTEGER NOT NULL,
    Spesa MONEY NOT NULL
);

--Aggiungo i vincoli:
--chiavi esterne:
--PartitaIva
ALTER TABLE Pubblicita ADD
CONSTRAINT PubblicitaSponsor FOREIGN KEY(PartitaIva)
REFERENCES SPONSOR(PartitaIva);
--CodConferenza

```

```

--se viene cancellata una conferenza, vengono cancellate
--anche tutte le pubblicita' legate a quella conferenza
ALTER TABLE Pubblicita ADD
CONSTRAINT PubblicitaConferenza FOREIGN KEY(CodConferenza)
REFERENCES CONFERENZA(CodConferenza) ON DELETE CASCADE;
--UnicoSponsor
--ogni coppia di sponsor e conferenza sponsorizzata e' unica
--e lo sponsor ha un'unica spesa per quella conferenza
ALTER TABLE Pubblicita ADD
CONSTRAINT UnicoSponsor UNIQUE(PartitaIva, CodConferenza);

```

4.2.15 Tabella AMMINISTRARE

```

CREATE TABLE AMMINISTRARE
(
NomeIstituzione VARCHAR(100) NOT NULL,
CodConferenza INTEGER NOT NULL
);

--Aggiungo i vincoli:
--chiavi esterne:
--NomeIstituzione
--se viene eliminata una conferenza, vengono cancellate
--tutte le amministrazioni (coppia conferenza-ente) di quella conferenza
ALTER TABLE AMMINISTRARE ADD
CONSTRAINT AmministrareConferenza FOREIGN KEY(CodConferenza)
REFERENCES CONFERENZA(CodConferenza) ON DELETE CASCADE;
--NomeIstituzione
ALTER TABLE AMMINISTRARE ADD
CONSTRAINT AmministrareEnte FOREIGN KEY(NomeIstituzione)
REFERENCES ENTE(NomeIstituzione);

```

4.2.16 Tabella ORGANIZZARE_L

```

--Creo la tabella Organizzare_L:
CREATE TABLE Organizzare_L
(
emailL VARCHAR(100) NOT NULL,
CodConferenza INTEGER NOT NULL
);

--Aggiungo i vincoli:
--chiavi esterne:
--CodConferenza
--se viene eliminata una conferenza, vengono cancellate
--anche le coppie di organizzatore-conferenza(cancellata)
ALTER TABLE ORGANIZZARE_L ADD
CONSTRAINT Organizzare_L_Conferenza FOREIGN KEY(CodConferenza)
REFERENCES CONFERENZA(CodConferenza) ON DELETE CASCADE;
--emailL
--se viene eliminato un Organizzatore Locale, vengono cancellate
--anche le coppie di organizzatore(cancellato)-conferenza
ALTER TABLE ORGANIZZARE_L ADD
CONSTRAINT Organizzatore_L FOREIGN KEY(emailL)
REFERENCES Organizzatore_Locale(emailL) ON DELETE CASCADE;

```

4.2.17 Tabella ORGANIZZARE_S

```
--Creo la tabella Organizzare_S
CREATE TABLE Organizzare_S
(
  emails VARCHAR(100) NOT NULL,
  CodConferenza INTEGER NOT NULL
);

--Aggiungo i vincoli:
--chiavi esterne:
--CodConferenza
--se viene eliminata una conferenza, vengono cancellate
--anche le coppie di organizzatore-conferenza(cancellata)
ALTER TABLE ORGANIZZARE_S ADD
CONSTRAINT Organizzare_S_Conferenza FOREIGN KEY(CodConferenza)
REFERENCES CONFERENZA(CodConferenza)ON DELETE CASCADE;
--emails
--se viene eliminato un Organizzatore Scientifico, vengono cancellate
--anche le coppie di organizzatore(cancellato)-conferenza
ALTER TABLE ORGANIZZARE_S ADD
CONSTRAINT Organizzatore_S FOREIGN KEY(emails)
REFERENCES Organizzatore_Scientifico(emails)ON DELETE CASCADE;
```

4.2.18 Tabella PARTECIPARE

```
--Creo la tabella PARTECIPARE
CREATE TABLE PARTECIPARE
(
  emailP VARCHAR(100) NOT NULL,
  CodSessione INTEGER NOT NULL
);

--Aggiungo i vincoli:
--chiavi esterne:
--CodSessione
--se viene eliminato un partecipante allora vengono eliminate
--le coppie sessione-partecipante del partecipante appena eliminato
ALTER TABLE PARTECIPARE ADD
CONSTRAINT ParteciparePartecipante FOREIGN KEY(emailP)
REFERENCES PARTECIPANTE(emailP)ON DELETE CASCADE;
--CodSessione
--se viene eliminata una sessione allora vengono eliminate
--le coppie sessione-partecipante della sessione appena eliminata
ALTER TABLE PARTECIPARE ADD
CONSTRAINT PartecipareSessione FOREIGN KEY(CodSessione)
REFERENCES SESSIONE(CodSessione)ON DELETE CASCADE;
```

4.3 Trigger, Procedure e View

4.3.1 AUTOMATISMO AMMINISTRARE ORGANIZZATORE

Per garantire che non esistano Organizzatori_locale¹ il cui ente non amministra la conferenza gestita dall'organizzatore, all'inserimento della coppia di PK dell'Organizzatore_locale e della conferenza che sta amministrando, nella tabella ponte Organizzare_L viene controllato se l'ente dell'organizzatore amministra la conferenza e in caso negativo vengono aggiunti nella tabella ponte l'Istituzione di appartenenza dell'organizzatore e il Cod della conferenza in questione. Automatizzando questo inserimento non è necessario quindi andare a lanciare alcuna eccezione che controlli la presenza dell'istituzione di un organizzatore_locale di una conferenza come sua amministratrice.

Struttura del trigger :

```
CREATE OR REPLACE FUNCTION Organizzare_L_Ente() RETURNS TRIGGER AS
$check$
DECLARE
    TempEnte ENTE.NomeIstituzione%TYPE;
    TempIstituzione ENTE.NomeIstituzione%TYPE;
BEGIN
    --estraggo l'istituzione di appartenenza dell'organizzatore
    SELECT Istituzione_di_Afferenza INTO TempIstituzione
    FROM ORGANIZZATORE_LOCALE      --della conferenza appena aggiunto
    WHERE emailL = NEW.emailL;
    --verifico se l'ente amministra già la conferenza
    SELECT NomeIstituzione FROM AMMINISTRARE AS A INTO TempEnte
    WHERE(A.NomeIstituzione = TempIstituzione AND
          A.CodConferenza = NEW.CodConferenza);

    --se non la amministra, allora aggiungo l'ente dell'organizzatore a quelli che
    amministrano la conferenza
    IF(TempEnte IS NULL) THEN
        INSERT INTO AMMINISTRARE(CodConferenza,NomeIstituzione)
        VALUES(NEW.CodConferenza, TempIstituzione);
    END IF;
    RETURN NEW;
END;
$check$ LANGUAGE PLPGSQL;

CREATE OR REPLACE TRIGGER Organizzare_L_Ente_Insert
BEFORE INSERT ON ORGANIZZARE_L
FOR EACH ROW
EXECUTE PROCEDURE Organizzare_L_Ente();
```

¹Lo stesso automatismo viene applicato per la tabella ponte Organizzare_S.

4.3.2 AUTOMATISMO PARTECIPAZIONE KEYNOTESPEAKER

Per garantire che non esistano KS che non partecipano a una sessione di cui sono KS, alla creazione di una sessione, nel caso in cui sia stato inserito un KS, questo viene automaticamente impostato come partecipante della sessione appena creata. La partecipazione è quindi garantita dall'insert nella tabella ponte della coppia di Primary Key di Sessione(appena creata) e di KeynoteSpeaker(della sessione appena creata). Automatizzando questo inserimento non è necessario quindi andare a lanciare alcuna eccezione che controlli la presenza del KS nella conferenza in cui interviene.

Struttura del trigger :

```
CREATE OR REPLACE FUNCTION KSPartecipante() RETURNS TRIGGER AS
$check$
DECLARE
    TempKS SESSION.CodSessione%TYPE;
BEGIN
    --controllo che sia stato realmente inserito un KS per la sessione
    IF(NEW.KeynoteSpeaker IS NOT NULL) THEN
        --controllo che nei partecipanti non ci sia già il KS della sessione
        SELECT CodSessione FROM PARTECIPARE AS P INTO TempKS
        WHERE (P.CodSessione = NEW.CodSessione AND
              P.emailP = NEW.KeynoteSpeaker);

        --se il KS non e' presente tra i partecipanti, allora lo inserisco
        IF(TempKS IS NULL) THEN
            INSERT INTO PARTECIPARE (CodSessione, emailP)
            VALUES(NEW.CodSessione,NEW.KeynoteSpeaker);
        END IF;
    END IF;
    RETURN NEW;
END;
$check$ LANGUAGE PLPGSQL;

CREATE OR REPLACE TRIGGER KSPartecipanteInsert
AFTER INSERT ON SESSIONE
FOR EACH ROW
EXECUTE PROCEDURE KSPartecipante();
```

4.3.3 TRIGGER CONFLISSO SESSIONI

All'inserimento di una sessione, nello specificare la locazione, gli orari di inizio e di fine è necessario andare a controllare che in quel determinato lasso di tempo (definito da orario inizio e fine) la locazione non sia già occupata da altre sessioni in corso. Il controllo del trigger avviene tramite una SELECT la quale estrae all'interno di TempSessione lo specifico codice della sessione che da Conflitto; va a ricercare quindi le sessioni che sono in corso tra orario inizio e orario fine appena inseriti, e che si svolgono nella locazione che è stata inserita. Nel caso in cui venga effettivamente trovata un CodSessione allora viene lanciata un'exception: 'Errore, in questa data e orario la locazione è occupata';

Definizione del trigger :

```
CREATE OR REPLACE FUNCTION ConflittoSessioni() RETURNS TRIGGER AS
$check$
DECLARE
    TempSessione SESSIONE.CodSessione%TYPE;
    DataSessione PROGRAMMA.DataProgramma%TYPE;
BEGIN
    --controllo che la locazione non sia già occupata a quella data e ora
    SELECT CodSessione FROM SESSIONE AS S
    INTO TempSessione
    WHERE S.NomeLocazione = NEW.NomeLocazione AND
           CodProgramma = NEW.CodProgramma AND
           ( (NEW.OrarioInizioSessione >= S.OrarioInizioSessione) AND
             (NEW.OrarioInizioSessione < S.OrarioFineSessione) ) OR
           ( (NEW.OrarioFineSessione > S.OrarioInizioSessione) AND (NEW.OrarioFineSessione
             <= S.OrarioFineSessione) ) );

    IF(TempSessione IS NOT NULL) THEN
        RAISE EXCEPTION 'Errore, in questa data e orario la locazione è occupata';
    END IF;
    RETURN NEW;
END;

$check$ LANGUAGE PLPGSQL;

--data una nuova sessione, bisogna controllare che questa
--possa essere ospitata in una locazione libera:
CREATE OR REPLACE TRIGGER ConflittoSessioniInsert
BEFORE INSERT ON SESSIONE
FOR EACH ROW
EXECUTE PROCEDURE ConflittoSessioni();
```

Esempio di attivazione del trigger :

```
--(per maggiore leggibilità non sarà riportata alcuna Descrizione della sessione).
INSERT INTO
    SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,DescrizioneSessione,
    Chair,KeynoteSpeaker,CodProgramma,NomeLocazione)
VALUES('9:00','10:30','Parts can worth like the whole','.',
    'FranciscoFlores-Revuelta@gmail.com','NicolaLerme@unipa.it',0,'Teatrino');

INSERT INTO
    SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,DescrizioneSessione,
    Chair,KeynoteSpeaker,CodProgramma,NomeLocazione)
VALUES('9:30','10:30','HBaXSCES - Human Behaviour Analysis for Smart City Environment
    Safety','.', 'JonAnderGomezAdrian@gmail.com','GiovanniRana@gmail.com',0,'Teatrino');
```

Output console :

```
ERROR:  Errore, in questa data e orario la locazione è occupata
CONTEXT:  funzione PL/pgSQL conflittosessioni() riga 14 a RAISE
SQL state: P0001
```

Tra le 9:30 e le 10:30 la locazione Teatrino è occupata dalla sessione "Parts can worth like the whole", non è quindi possibile aggiungere la sessione "HBAXSCES" in Teatrino; verrà quindi lanciata l'exception.

4.3.4 TRIGGER LOCAZIONE VALIDA

All'inserimento di una sessione, nello specificare la locazione in cui questa si svolga è necessario controllare che questa sia una delle locazioni interne alla Sede della conferenza che ospita la sessione appena creata. Il controllo del trigger avviene tramite una prima SELECT la quale estrae il nome della sede in cui si svolge la conferenza della sessione creata. Una seconda SELECT estrae la locazione che abbia la stessa PK della locazione della sessione inserita ("NEW.NomeLocazione = NomeLocazione") e che appartenga alla sede estratta dalla precedente SELECT. Nel caso in cui non viene estratta una locazione dalla seconda SELECT, vuol dire che questa non si trova all'interno della sede in cui viene svolta la conferenza della sessione appena inserita. Viene dunque chiamata una exception: 'Errore, in questa sede non esiste questa locazione'.

Definizione del trigger :

```
CREATE OR REPLACE FUNCTION LocazioneValida() RETURNS TRIGGER AS
$check$
DECLARE
    TempSede SEDE.NomeSede%TYPE;
    TempLocazione LOCAZIONE.NomeLocazione%TYPE;
BEGIN
    --estraggo la sede della sessione che sto inserendo
    SELECT NomeSede FROM ((PROGRAMMA NATURAL JOIN CONFERENZA) NATURAL JOIN SEDE)
    INTO TempSede
    WHERE (NEW.CodProgramma = CodProgramma);

    --estraggo la locazione che potrebbe dare conflitto
    SELECT NomeLocazione FROM (LOCAZIONE NATURAL JOIN SEDE)
    INTO TempLocazione
    WHERE (NomeSede = TempSede AND NEW.NomeLocazione = NomeLocazione);

    --se non trovo alcuna locazione, allora questa non esiste in quella sede
    IF (TempLocazione IS NULL) THEN
        RAISE EXCEPTION 'Errore, in questa conferenza non esiste questa locazione';
    END IF;
    RETURN NEW;
END;
$check$ LANGUAGE PLPGSQL;

CREATE OR REPLACE TRIGGER LocazioneValidaInsert
BEFORE INSERT ON SESSIONE
FOR EACH ROW
EXECUTE PROCEDURE LocazioneValida();
```

Esempio di attivazione del trigger :

```
INSERT INTO LOCAZIONE (NomeLocazione, NomeSede)
VALUES('Sala sul Chiostro del 500','convitto palmieri');

--(per maggiore leggibilita non sara riportata alcuna Descrizione della conferenza).
INSERT INTO CONFERENZA(CodConferenza,
    TitoloConferenza,DataInizio,DataFine,Descrizione,NomeSede)
VALUES('IEEE CSR CyberSecurity and Resilience', '27/07/2022','29/07/2022','.', 'monte
    sant'angelo');

INSERT INTO PROGRAMMA(CodProgramma, DataProgramma,CodConferenza)
VALUES(5, '27/07/2022',2);

INSERT INTO
    SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,Chair,CodProgramma,
    NomeLocazione)
VALUES('9:00','10:00','Plenary session
    PL-1','FranciscoFlorez-Revuelta@gmail.com',5, 'Sala sul Chiostro del 500 (First
    floor)');
```

Output console :

```
ERROR: Errore, in questa conferenza non esiste questa locazione
CONTEXT: funzione PL/pgSQL locazionevalida() riga 15 a RAISE
SQL state: P0001
```

La locazione Sala sul Chiostro del 500 (First floor) appartiene quindi alla sede Convitto Palmieri; la sessione (avente come locazione Sala sul Chiostro del 500 (First floor)) appartiene a una conferenza che ha come sede monte sant'angelo (che NON è la sede della Sala sul chiostro del 500), quindi viene lanciata l'exception.

4.4 TRIGGER CONFLITTO CONFERENZE

All'inserimento di una conferenza, nello specificare la data di inizio e di fine è necessario andare a controllare che in quel lasso di tempo(definito da data inizio e fine) la sede non sia già occupata da altre conferenze in corso. Il controllo del trigger avviene tramite una SELECT la quale estrae all'interno di TempConferenza lo specifico codice della conferenza che da conflitto; va a ricercare quindi le conferenze che sono in corso tra data inizio e data fine(compresi)appena inseriti, e che si svolgono nella sede inserita. Nel caso in cui venga effettivamente trovato un CodConferenza allora viene lanciata un'exception: 'Errore, la sede ospita già una conferenza';

Definizione del trigger :

```
CREATE OR REPLACE FUNCTION NoConferenzaParallela() RETURNS TRIGGER AS
$check$
DECLARE
    TempConferenza CONFERENZA.CodConferenza%TYPE;
BEGIN
    --controllo che la nuova conferenza inserita(o updatata) non abbia la sede occupata
    in quella data
    SELECT CodConferenza FROM CONFERENZA INTO TempConferenza
    WHERE NomeSede = NEW.NomeSede AND
        ((NEW.DataInizio BETWEEN DataInizio AND DataFine) OR
        (NEW.DataFine BETWEEN DataInizio AND DataFine));

    IF(TempConferenza IS NOT NULL) THEN
        RAISE EXCEPTION 'Errore, la sede ospita gia una conferenza';
    END IF;
    RETURN NEW;
END;
$check$ LANGUAGE PLPGSQL;
```



```
--data una nuova conferenza, bisogna controllare che questa
--possa essere ospitata in una sede libera(in inserimento)
CREATE OR REPLACE TRIGGER NoConferenzaParallelaInsert
BEFORE INSERT ON CONFERENZA
FOR EACH ROW
EXECUTE PROCEDURE NoConferenzaParallela();
```

Esempio di attivazione del trigger :

```
INSERT INTO CONFERENZA(TitoloConferenza,DataInizio,DataFine,Descrizione,NomeSede)
VALUES('21st International Conference on Image Analysis and Processing',
      '23/05/2022','27/05/2022',
      '.', 'monte sant''angelo');

--(per maggiore leggibilita' non sara' riportata alcuna Descrizione della conferenza).
INSERT INTO CONFERENZA(TitoloConferenza,DataInizio,DataFine,Descrizione,NomeSede)
VALUES('IEEE CSR CyberSecurity and Resilience', '26/05/2022','29/05/2022',
      '.', 'monte sant''angelo');
```

Output console :

```
ERROR:  Errore, la sede ospita gia una conferenza
CONTEXT:  funzione PL/pgSQL noconferenzaparallela() riga 11 a RAISE
SQL state: P0001
```

Nella sede monte sant'angelo, nel giorno 26/05/2022 si svolgera' la conferenza '21st International...', non e' quindi possibile aggiungere la conferenza 'IEEE CSR CyberSecurity...' avente come data di svolgimento 26/05/2022 e come locazione monte sant'angelo; viene dunque lanciata l'exception.

4.4.1 VIEW RIEPILOGOKS

Questa view ci permette di ottenere un riepilogo annuale sul numero di KeynoteSpeaker per ogni Istituzione di Afferenza che ha partecipato a organizzazione di conferenze in quell'anno.

Struttura view :

```
CREATE VIEW RiepilogoKS AS
SELECT istituzione_di_afferenza, count(keynoteSpeaker)
FROM PARTECIPANTE NATURAL JOIN(
  --selezione in modo distinto tutti i KS dell'anno
  SELECT DISTINCT KeynoteSpeaker FROM SESSIONE NATURAL JOIN PROGRAMMA
  WHERE extract (year from DataProgramma) = extract(year from CURRENT_DATE)) AS T
WHERE emailp = KeynoteSpeaker
GROUP BY istituzione_di_afferenza;
```

Esempio di attivazione del trigger :

```
SELECT * FROM RiepilogoKS;
```



	istituzione_di_afferenza 	count 
	character varying (100)	bigint
1	Bournemouth University	1
2	Università degli Studi di...	2
3	Università del Salento	1
4	Università di Pavia	1

Figura 1: View in pgAdmin

4.5 Popolamento del database

```
--popolo la tabella SEDE:
INSERT INTO SEDE(NomeSede,NomeVia,NumeroCivico,Citta)
VALUES('convitto palmieri','piazzetta carducci',28,'lecce');

INSERT INTO SEDE(NomeSede,NomeVia,NumeroCivico,Citta)
VALUES('monte sant''angelo','via cintia',21,'napoli');

INSERT INTO SEDE(NomeSede,NomeVia,NumeroCivico,Citta)
VALUES('porta di massa','via porta di massa',1,'napoli');


--popolo la tabella LOCAZIONE:
INSERT INTO LOCAZIONE (NomeLocazione, NomeSede)
VALUES('Athena','monte sant''angelo');

INSERT INTO LOCAZIONE (NomeLocazione, NomeSede)
VALUES('Nafsika','monte sant''angelo');

INSERT INTO LOCAZIONE (NomeLocazione, NomeSede)
VALUES('Sala d''arte (First floor)','convitto palmieri');

INSERT INTO LOCAZIONE (NomeLocazione, NomeSede)
VALUES('Sala Museo della Stampa','convitto palmieri');

INSERT INTO LOCAZIONE (NomeLocazione, NomeSede)
VALUES('Teatrino','convitto palmieri');

INSERT INTO LOCAZIONE (NomeLocazione, NomeSede)
VALUES('Sala sul Chiostro del 500 (First floor)','convitto palmieri');

INSERT INTO LOCAZIONE (NomeLocazione, NomeSede)
VALUES('Chiostro del 500','convitto palmieri');

INSERT INTO LOCAZIONE (NomeLocazione, NomeSede)
VALUES('AulaP','porta di massa');

INSERT INTO LOCAZIONE (NomeLocazione, NomeSede)
VALUES('Sala Grande','porta di massa');


--popolo la tabella CONFERENZA:
INSERT INTO CONFERENZA(TitoloConferenza,DataInizio,DataFine,Descrizione,NomeSede)
VALUES('21st International Conference on Image Analysis and Processing',
      '23/05/2022','27/05/2022',
      'ICIAP: 21st International Conference on image analysis and processing 24 May 2022
      Vi aspettiamo a ICIAP, la Conferenza organizzata ogni due anni dall''Association
      for
      Research in Computer Vision, Pattern Recognition and Machine Learning (CVPL, ex
      GIRPR) che e'' parte dell''International Association for Pattern Recognition
      (IAPR).','convitto palmieri');
```

```

INSERT INTO CONFERENZA(TitoloConferenza,DataInizio,DataFine,Descrizione,NomeSede)
VALUES('IEEE CSR CyberSecurity and Resilience', '27/07/2022','29/07/2022',
      'The IEEE International Conference on Cyber Security and Resilience recognizes
      outstanding individuals
      who make substantial contributions to the advancement of
      security and resilience, inspiring other members of the community with their
      pioneering
      research and innovation. The awardees need not be IEEE members.','monte
      sant''angelo');

--popolo la tabella SPONSOR:
INSERT INTO SPONSOR(PartitaIVA,NomeAzienda)
VALUES('04935230963','Apple Retail Italia S.r.l.');
```

```

INSERT INTO SPONSOR(PartitaIVA,NomeAzienda)
VALUES('11325690151','Samsung Electronics Italia S.p.A.');
```

```

INSERT INTO SPONSOR(PartitaIVA,NomeAzienda)
VALUES('00348270133','META S.P.A.');
```

```

--popolo la tabella PROGRAMMA:
INSERT INTO PROGRAMMA(DataProgramma,CodConferenza)
VALUES('23/05/2022',1);

INSERT INTO PROGRAMMA(DataProgramma,CodConferenza)
VALUES('24/05/2022',1);

INSERT INTO PROGRAMMA(DataProgramma,CodConferenza)
VALUES('25/05/2022',1);

INSERT INTO PROGRAMMA(DataProgramma,CodConferenza)
VALUES('26/05/2022',1);

INSERT INTO PROGRAMMA(DataProgramma,CodConferenza)
VALUES('27/05/2022',1);

INSERT INTO PROGRAMMA(DataProgramma,CodConferenza)
VALUES('27/07/2022',2);

INSERT INTO PROGRAMMA(DataProgramma,CodConferenza)
VALUES('28/07/2022',2);

INSERT INTO PROGRAMMA(DataProgramma,CodConferenza)
VALUES('29/07/2022',2);

--popolo la tabella INTERVALLO:
INSERT INTO
      INTERVALLO(TipoIntervallo,OrarioInizioIntervallo,OrarioFineIntervallo,CodProgramma)
VALUES('CoffeBreak', '10:30:00','11:00:00',0);
```

```

INSERT INTO
    INTERVALLO (TipoIntervallo, OrarioInizioIntervallo, OrarioFineIntervallo, CodProgramma)
VALUES ('CoffeBreak', '15:30:00', '16:00:00', 0);

INSERT INTO
    INTERVALLO (TipoIntervallo, OrarioInizioIntervallo, OrarioFineIntervallo, CodProgramma)
VALUES ('CoffeBreak', '10:30:00', '11:00:00', 1);

INSERT INTO
    INTERVALLO (TipoIntervallo, OrarioInizioIntervallo, OrarioFineIntervallo, CodProgramma)
VALUES ('Pranzo', '13:00:00', '14:00:00', 1);

INSERT INTO
    INTERVALLO (TipoIntervallo, OrarioInizioIntervallo, OrarioFineIntervallo, CodProgramma)
VALUES ('CoffeBreak', '15:30:00', '16:00:00', 1);

INSERT INTO
    INTERVALLO (TipoIntervallo, OrarioInizioIntervallo, OrarioFineIntervallo, CodProgramma)
VALUES ('CoffeBreak', '10:30:00', '11:00:00', 2);

INSERT INTO
    INTERVALLO (TipoIntervallo, OrarioInizioIntervallo, OrarioFineIntervallo, CodProgramma)
VALUES ('Pranzo', '13:00:00', '14:00:00', 2);

INSERT INTO
    INTERVALLO (TipoIntervallo, OrarioInizioIntervallo, OrarioFineIntervallo, CodProgramma)
VALUES ('CoffeBreak', '10:30:00', '11:00:00', 3);

INSERT INTO
    INTERVALLO (TipoIntervallo, OrarioInizioIntervallo, OrarioFineIntervallo, CodProgramma)
VALUES ('Pranzo', '13:00:00', '14:00:00', 3);

INSERT INTO
    INTERVALLO (TipoIntervallo, OrarioInizioIntervallo, OrarioFineIntervallo, CodProgramma)
VALUES ('CoffeBreak', '15:30:00', '16:00:00', 3);

INSERT INTO
    INTERVALLO (TipoIntervallo, OrarioInizioIntervallo, OrarioFineIntervallo, CodProgramma)
VALUES ('CoffeBreak', '10:30:00', '11:00:00', 4);

INSERT INTO
    INTERVALLO (TipoIntervallo, OrarioInizioIntervallo, OrarioFineIntervallo, CodProgramma)
VALUES ('Pranzo', '13:00:00', '14:00:00', 4);

INSERT INTO
    INTERVALLO (TipoIntervallo, OrarioInizioIntervallo, OrarioFineIntervallo, CodProgramma)
VALUES ('CoffeBreak', '15:30:00', '16:00:00', 4);

INSERT INTO
    INTERVALLO (TipoIntervallo, OrarioInizioIntervallo, OrarioFineIntervallo, CodProgramma)
VALUES ('CoffeBreak', '11:40:00', '12:00:00', 5);

INSERT INTO
    INTERVALLO (TipoIntervallo, OrarioInizioIntervallo, OrarioFineIntervallo, CodProgramma)
VALUES ('Pranzo', '13:20:00', '14:20:00', 5);

INSERT INTO
    INTERVALLO (TipoIntervallo, OrarioInizioIntervallo, OrarioFineIntervallo, CodProgramma)
VALUES ('CoffeBreak', '16:00:00', '16:20:00', 5);

INSERT INTO
    INTERVALLO (TipoIntervallo, OrarioInizioIntervallo, OrarioFineIntervallo, CodProgramma)

```

```

VALUES('CoffeBreak','11:40:00','12:00:00',6);

INSERT INTO
INTERVALLO(TipoIntervallo,OrarioInizioIntervallo,OrarioFineIntervallo,CodProgramma)
VALUES('Pranzo','13:20:00','14:20:00',6);

INSERT INTO
INTERVALLO(TipoIntervallo,OrarioInizioIntervallo,OrarioFineIntervallo,CodProgramma)
VALUES('CoffeBreak','18:00:00','18:20:00',6);

INSERT INTO
INTERVALLO(TipoIntervallo,OrarioInizioIntervallo,OrarioFineIntervallo,CodProgramma)
VALUES('CoffeBreak','11:40:00','12:00:00',7);

INSERT INTO
INTERVALLO(TipoIntervallo,OrarioInizioIntervallo,OrarioFineIntervallo,CodProgramma)
VALUES('Pranzo','13:20:00','14:20:00',7);

--popolo la tabella EVENTO_SOCIALE:
INSERT INTO EVENTO_SOCIALE(TipoEvento,OrarioInizioEvento,OrarioFineEvento,CodProgramma)
VALUES('Cena','19:30:00','22:00:00',2);

INSERT INTO EVENTO_SOCIALE(TipoEvento,OrarioInizioEvento,OrarioFineEvento,CodProgramma)
VALUES('Cena','18:50:00','23:30:00',3);

INSERT INTO EVENTO_SOCIALE(TipoEvento,OrarioInizioEvento,OrarioFineEvento,CodProgramma)
VALUES('Gita','18:50:00','23:30:00',4);

--popolo la tabella ENTE:
INSERT INTO ENTE(NomeIstituzione)
VALUES('Universita degli Studi di Napoli Federico II');

INSERT INTO ENTE(NomeIstituzione)
VALUES('Universita Parthenope');

INSERT INTO ENTE(NomeIstituzione)
VALUES('Universita del Salento');

INSERT INTO ENTE(NomeIstituzione)
VALUES('Universita Vanvitelli');

INSERT INTO ENTE(NomeIstituzione)
VALUES('Universita degli Studi di Salerno');

INSERT INTO ENTE(NomeIstituzione)
VALUES('University of Cambridge');

INSERT INTO ENTE(NomeIstituzione)
VALUES('Universita di Pavia');

INSERT INTO ENTE(NomeIstituzione)
VALUES('Bournemouth University');

INSERT INTO ENTE(NomeIstituzione)

```

```

VALUES('Humanitas University');

INSERT INTO ENTE(NomeIstituzione)
VALUES('University of Zilina');

INSERT INTO ENTE(NomeIstituzione)
VALUES('University of Alicante');

INSERT INTO ENTE(NomeIstituzione)
VALUES('Universita di Catania');


--popolo la tabella ORGANIZZATORE_LOCALE:
INSERT INTO
    ORGANIZZATORE_LOCALE(emailL,password,Titolo,Nome,Cognome,Istituzione_di_Afferenza)
VALUES('CarmenBisogni@unisa.it','Conferenze1000','Mss', 'Carmen','Bisogni', 'Universita
degli Studi di Salerno');

INSERT INTO
    ORGANIZZATORE_LOCALE(emailL,password,Titolo,Nome,Cognome,Istituzione_di_Afferenza)
VALUES('PiercarloDondi@unipa.it','Conferenze1000','Professor', 'Piercarlo','Dondi',
'Universita di Pavia');

INSERT INTO
    ORGANIZZATORE_LOCALE(emailL,password,Titolo,Nome,Cognome,Istituzione_di_Afferenza)
VALUES('FabioNarducci@unisa.it','Conferenze1000','Mr', 'Fabio','Narducci', 'Universita
degli Studi di Salerno');

INSERT INTO
    ORGANIZZATORE_LOCALE(emailL,password,Titolo,Nome,Cognome,Istituzione_di_Afferenza)
VALUES('AlessandroBruno@gmail.it','Conferenze1000','Mr', 'Alessandro','Bruno',
'Humanitas University');

INSERT INTO
    ORGANIZZATORE_LOCALE(emailL,password,Titolo,Nome,Cognome,Istituzione_di_Afferenza)
VALUES('SimonePalazzo@unica.it','Conferenze1000','Mr', 'Simone','Palazzo', 'Universita
di Catania');

INSERT INTO
    ORGANIZZATORE_LOCALE(emailL,password,Titolo,Nome,Cognome,Istituzione_di_Afferenza)
VALUES('FedericaProiettoSalanitri@unisa.it','Conferenze1000','Mss', 'Federica','Proietto
Salanitri', 'Universita di Catania');

INSERT INTO
    ORGANIZZATORE_LOCALE(emailL,password,Titolo,Nome,Cognome,Istituzione_di_Afferenza)
VALUES('VincenzoLomonaco@unipa.it','Conferenze1000','Professor', 'Vincenzo','Lomonaco',
'Universita di Pavia');


--popolo la tabella ORGANIZZATORE_SCIENTIFICO:
INSERT INTO ORGANIZZATORE_SCIENTIFICO(emailS,password,Titolo,Nome,Cognome,
Istituzione_di_Afferenza)
VALUES('ZoheirSabeur@gmail.com','Conferenze1000','Professor', 'Zoheir','Sabeur',
'Bournemouth University');

```

```

INSERT INTO ORGANIZZATORE_SCIENTIFICO(emailS,password,Titolo,Nome,Cognome,
Istituzione_di_Afferenza)
VALUES('DenizChetinkaya@gmail.com','Conferenze1000','Professoressa',
'Deniz','Chetinkaya','Bournemouth University');

INSERT INTO ORGANIZZATORE_SCIENTIFICO(emailS,password,Titolo,Nome,Cognome,
Istituzione_di_Afferenza)
VALUES('FranciscoFlorez-Revuelta@gmail.com','Conferenze1000','Mr',
'Francisco','Florez-Revuelta','University of Alicante');

INSERT INTO ORGANIZZATORE_SCIENTIFICO(emailS,password,Titolo,Nome,Cognome,
Istituzione_di_Afferenza)
VALUES('PeterPocota@gmail.com','Conferenze1000','Mr','Peter','Pocota','University of
Zilina');

INSERT INTO ORGANIZZATORE_SCIENTIFICO(emailS,password,Titolo,Nome,Cognome,
Istituzione_di_Afferenza)
VALUES('JonAnderGomezAdrian@gmail.com','Conferenze1000','Mr','Jon Ander','Gomez
Adrian','University of Alicante');

INSERT INTO ORGANIZZATORE_SCIENTIFICO(emailS,password,Titolo,Nome,Cognome,
Istituzione_di_Afferenza)
VALUES('StephenHawking@gmail.com','Conferenze1000','Mr','Stephen','Hawking',
'University of Cambridge');

--popolo la tabella PARTECIPANTE:
INSERT INTO PARTECIPANTE(emailP, Titolo, Nome, Cognome, Istituzione_di_Afferenza)
VALUES('NicolaLerme@unipa.it','Mr','Nicola','Lerme','Universita di Pavia');

INSERT INTO PARTECIPANTE(emailP, Titolo, Nome, Cognome, Istituzione_di_Afferenza)
VALUES('LorenzoBaraldi@unina.it','Mr','Lorenzo','Baraldi','Universita degli Studi di
Napoli Federico II');

INSERT INTO PARTECIPANTE(emailP, Titolo, Nome, Cognome, Istituzione_di_Afferenza)
VALUES('AlessandroGrieco@unina.it','Mr','Alessandro','Grieco','Universita degli Studi di
Napoli Federico II');

INSERT INTO PARTECIPANTE(emailP, Titolo, Nome, Cognome, Istituzione_di_Afferenza)
VALUES('StefanoAllegretti@unipart.it','Mr','Stefano','Allegretti','Universita
Parthenope');

INSERT INTO PARTECIPANTE(emailP, Titolo, Nome, Cognome, Istituzione_di_Afferenza)
VALUES('DarioMoccia@unipart.it','Professor','Dario','Moccia','Universita Parthenope');

INSERT INTO PARTECIPANTE(emailP, Titolo, Nome, Cognome, Istituzione_di_Afferenza)
VALUES('ClaudioFerrari@unica.it','Professor','Claudio','Ferrari','Universita di
Catania');

INSERT INTO PARTECIPANTE(emailP, Titolo, Nome, Cognome, Istituzione_di_Afferenza)
VALUES('FedericoBeccaria@univa.it','Mr','Federico','Beccaria','Universita Vanvitelli');

INSERT INTO PARTECIPANTE(emailP, Titolo, Nome, Cognome, Istituzione_di_Afferenza)
VALUES('AndreaPilzer@univa.it','Professor','Andrea','Pizler','Universita Vanvitelli');

INSERT INTO PARTECIPANTE(emailP, Titolo, Nome, Cognome, Istituzione_di_Afferenza)

```

```

VALUES('GiovanniRana@gmail.com','Professor','Giovanni','Rana','Bournemouth University');

INSERT INTO PARTECIPANTE(emailP, Titolo, Nome, Cognome, Istituzione_di_Afferenza)
VALUES('CosimoDistante@unisal.it','Professor','Cosimo','Distante','Universita del
      Salento');

INSERT INTO PARTECIPANTE(emailP, Titolo, Nome, Cognome, Istituzione_di_Afferenza)
VALUES('FrancescoBarecchia@unina.it','Mr','Francesco','Barecchia','Universita degli
      Studi di Napoli Federico II');

--popolo la tabella ponte ORGANIZZARE_S:
INSERT INTO ORGANIZZARE_S(CodConferenza,emailS)
VALUES(1,'FranciscoFlorez-Revuelta@gmail.com');

INSERT INTO ORGANIZZARE_S(CodConferenza,emailS)
VALUES(1,'DenizChetinkaya@gmail.com');

INSERT INTO ORGANIZZARE_S(CodConferenza,emailS)
VALUES(1,'JonAnderGomezAdrian@gmail.com');

INSERT INTO ORGANIZZARE_S(CodConferenza,emailS)
VALUES(1,'PeterPocta@gmail.com');

INSERT INTO ORGANIZZARE_S(CodConferenza,emailS)
VALUES(1,'ZoheirSabeur@gmail.com');

INSERT INTO ORGANIZZARE_S(CodConferenza,emailS)
VALUES(1,'StephenHawking@gmail.com');

INSERT INTO ORGANIZZARE_S(CodConferenza,emailS)
VALUES(2,'FranciscoFlorez-Revuelta@gmail.com');

INSERT INTO ORGANIZZARE_S(CodConferenza,emailS)
VALUES(2,'DenizChetinkaya@gmail.com');

--popolo la tabella SESSIONE:
INSERT INTO
      SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,DescrizioneSessione
      ,Chair,KeynoteSpeaker,CodProgramma,NomeLocazione)
VALUES('9:00:00','10:30:00','Parts can worth like the whole','Quite often the useful
      data for an analysis task are not available in an optimal condition. This may be
      due to the occlusions or the noise affecting the
      acquisition of the samples, or in some cases the problem itself is conceived in a way
      that a solution comes from the analysis of smaller portions of the input.',
      'FranciscoFlorez-Revuelta@gmail.com','NicolaLerme@unipa.it',0,'Sala sul Chiostro del 500
      (First floor)');

INSERT INTO
      SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,DescrizioneSessione,
      Chair,KeynoteSpeaker,CodProgramma,NomeLocazione)
VALUES('11:00','15:30','HBAXSCES - Human Behaviour Analysis for Smart City Environment
      Safety','Nowadays, Smart Cities aim to ensure secure and safe physical and digital
      environments for the well-being of citizens.
      Among many, ICT systems are reliant on evolving Artificial Intelligence, Pattern

```



```

Recognition, Computer Vision, 3D simulations and Digital Twins techniques to make
the environments more resilient.',
'JonAnderGomezAdrian@gmail.com','GiovanniRana@gmail.com',0,'Teatrino');

INSERT INTO
SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,DescrizioneSessione,
Chair,CodProgramma,NomeLocazione)
VALUES('16:00','18:00','Deep Learning High Performance Computing to Boost Biomedical
Applications','Healthcare is one of the key sectors of the global economy,
especially in Europe. Any improvement in healthcare
systems has a high impact on the welfare of the society. The use of technologies in
health is clearly a strong path to more efficient healthcare, benefitting both
individual people and the publicbudgets. European
public health systems are generating large datasets of biomedical data in general, and
images in particular, as most medical examinations use image-based processes.',
'PeterPocota@gmail.com',0,'Sala d''arte (First floor)');

INSERT INTO
SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,DescrizioneSessione,
Chair,KeynoteSpeaker,CodProgramma,NomeLocazione)
VALUES('9:00','10:30','Deep Learning for Visual Object Tracking Pt1','In its simplest
definition, visual object tracking consists in the persistent recognition and
localization of a generic target object in a video.
Several challenges such as object occlusions, pose and scale changes, rotations and
shape variations, and the presence of similar objects, must be tackled to
accurately keep track of a target''s position. The ultimate
goal of generic object tracking is to build robust models capable to overcome such
challenging factors. In the past, such issues have been addressed by disparate
principles formalizing the concepts of appearance
model, motion model, and matching operation. In recent years, algorithms based on deep
learning tried to learn such conceptual blocks by exploiting the ability of deep
neural networks in learning complex functions
from visual
examples.','ZoheirSabeur@gmail.com','AlessandroGrieco@unina.it',1,'Teatrino');

INSERT INTO
SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,DescrizioneSessione,
Chair,KeynoteSpeaker,CodProgramma,NomeLocazione)
VALUES('11:00','13:00','Deep Learning for Visual Object Tracking Pt2','In its simplest
definition, visual object tracking consists in the persistent recognition and
localization of a generic target object in a video.
Several challenges such as object occlusions, pose and scale changes, rotations and
shape variations, and the presence of similar objects, must be tackled to
accurately keep track of a target''s position. The ultimate
goal of generic object tracking is to build robust models capable to overcome such
challenging factors. In the past, such issues have been addressed by disparate
principles formalizing the concepts of appearance model,
motion model, and matching operation. In recent years, algorithms based on deep learning
tried to learn such conceptual blocks by exploiting the ability of deep neural
networks in learning complex functions from visual
examples.','ZoheirSabeur@gmail.com','AlessandroGrieco@unina.it',1,'Teatrino');

INSERT INTO
SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,DescrizioneSessione,
Chair,CodProgramma,NomeLocazione)
VALUES('11:00','13:00','WOSDETC - Small-Drone Surveillance, Detection and Counteraction
Techniques','In the last few years the popularity of small Remotely Piloted
Aircraft Systems (RPAS) and more generally (also
autonomous) "drones", has exponentially increased due to the availability of low-cost
off-the-shelf products, including build-from-scratch and DIY kits. At the same
time, issues regarding safety, privacy and security
aspects are arising. There is inded a gap in current surveillance systems for the

```

```

        detection of such flying systems, which can be used for illegal activities such as
        smuggling of drugs or espionage, as well as for
        carrying explosives or chemical weapons.', 'PeterPocota@gmail.com', 1, 'Sala sul Chiostro
        del 500 (First floor)');

INSERT INTO
    SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,DescrizioneSessione,
    Chair,KeynoteSpeaker,CodProgramma,NomeLocazione)
VALUES('14:00','15:30','DEEP LEARNING FOR MULTI-GPUS','Deep Learning has been the most
    significant breakthrough in the past 10 years in the field of pattern recognition
    and machine learning. It has achieved significant
    advancements in terms of the effectiveness of prediction models on many research topics
    and application fields, ranging from computer vision, natural language processing,
    embodied AI and to more traditional fields of
    pattern recognition. This paradigm shift has radically changed the research methodology
    towards a data-oriented approach, in which learning involves all steps of the
    prediction pipeline from feature extraction to
    classification.', 'JonAnderGomezAdrian@gmail.com', 'FrancescoBarecchia@unina.it', 1,
    'Teatrino');

INSERT INTO SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,
    Chair,CodProgramma,NomeLocazione)
VALUES('8:30','10:30','Oral Session 1: Image Analysis, Detection and
    Recognition','ZoheirSabeur@gmail.com', 2, 'Sala d''arte (First floor)');

INSERT INTO SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,
    Chair,CodProgramma,NomeLocazione)
VALUES('14:00','15:30','Message from the General
    Chair','DenizChetinkaya@gmail.com', 2, 'Teatrino');

INSERT INTO
    SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,DescrizioneSessione,
    Chair,KeynoteSpeaker,CodProgramma,NomeLocazione)
VALUES('11:00','13:00','Industrial Panel','The Industrial Panel session will start with
    presentations by the panelists, introducing the companies and the technological
    challenges of their business.',
    'PeterPocota@gmail.com', 'CosimoDistante@unisal.it', 3, 'Sala Museo della Stampa');

INSERT INTO SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,
    Chair,CodProgramma,NomeLocazione)
VALUES('14:00','15:30','Shifting paradigms in multi-object
    tracking','FranciscoFlores-Revuelta@gmail.com', 3, 'Sala Museo della Stampa');

INSERT INTO
    SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,DescrizioneSessione,
    Chair,KeynoteSpeaker,CodProgramma,NomeLocazione)
VALUES('11:00','13:00','Video Understanding - An Egocentric Perspective','This talk aims
    to argue for a fine(r)-grained perspective onto human-object interactions, from
    video sequences, captured
    in an egocentric perspective (i.e. first-person footage).',
    'FranciscoFlores-Revuelta@gmail.com', 'NicolaLerme@unipa.it', 4, 'Sala d''arte (First
    floor)');

INSERT INTO SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,
    Chair,CodProgramma,NomeLocazione)
VALUES('14:00','15:30','Closing remarks','StephenHawking@gmail.com', 4, 'Sala Museo della
    Stampa');

INSERT INTO SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,
    Chair,CodProgramma,NomeLocazione)
VALUES('9:00','10:00','Plenary session
    PL-1','FranciscoFlores-Revuelta@gmail.com', 5, 'Athena');

```

```

INSERT INTO SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,
Chair,CodProgramma,NomeLocazione)
VALUES('10:00','11:40','CSR-1','FranciscoFlorez-Revuelta@gmail.com',5,'Athena');

INSERT INTO SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,
Chair,CodProgramma,NomeLocazione)
VALUES('12:00','13:20','CSR-2','FranciscoFlorez-Revuelta@gmail.com',5,'Athena');

INSERT INTO SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,
Chair,CodProgramma,NomeLocazione)
VALUES('14:20','16:00','CSR-3','FranciscoFlorez-Revuelta@gmail.com',5,'Athena');

INSERT INTO SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,
Chair,CodProgramma,NomeLocazione)
VALUES('16:20','18:00','CSR-2','FranciscoFlorez-Revuelta@gmail.com',5,'Athena');

INSERT INTO SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,
Chair,CodProgramma,NomeLocazione)
VALUES('10:00','11:40','WS-DS4CS-1','DenizChetinkaya@gmail.com',5,'Nafsika');

INSERT INTO SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,
Chair,CodProgramma,NomeLocazione)
VALUES('12:00','13:20','CSR-2','DenizChetinkaya@gmail.com',5,'Nafsika');

INSERT INTO SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,
Chair,CodProgramma,NomeLocazione)
VALUES('15:00','16:00','CSR-3','DenizChetinkaya@gmail.com',5,'Nafsika');

INSERT INTO SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,
Chair,CodProgramma,NomeLocazione)
VALUES('16:20','18:00','CSR-2','DenizChetinkaya@gmail.com',5,'Nafsika');

INSERT INTO SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,
Chair,CodProgramma,NomeLocazione)
VALUES('9:00','10:00','PL-2','FranciscoFlorez-Revuelta@gmail.com',6,'Athena');

INSERT INTO SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,
Chair,CodProgramma,NomeLocazione)
VALUES('10:00','11:40','CSR-5','FranciscoFlorez-Revuelta@gmail.com',6,'Athena');

INSERT INTO SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,
Chair,CodProgramma,NomeLocazione)
VALUES('12:00','13:20','CSR-6','FranciscoFlorez-Revuelta@gmail.com',6,'Athena');

INSERT INTO SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,
Chair,CodProgramma,NomeLocazione)
VALUES('14:20','15:20','PL-3','FranciscoFlorez-Revuelta@gmail.com',6,'Athena');

INSERT INTO SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,
Chair,CodProgramma,NomeLocazione)
VALUES('15:20','16:20','AW','FranciscoFlorez-Revuelta@gmail.com',6,'Athena');

INSERT INTO SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,
Chair,CodProgramma,NomeLocazione)
VALUES('10:00','11:40','WS-CRE-1','DenizChetinkaya@gmail.com',6,'Nafsika');

INSERT INTO SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,
Chair,CodProgramma,NomeLocazione)
VALUES('12:00','13:20','WS-CRST','DenizChetinkaya@gmail.com',6,'Nafsika');

INSERT INTO SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,

```

```

Chair,CodProgramma,NomeLocazione)
VALUES('16:20','18:00','WS-CRE-2','DenizChetinkaya@gmail.com',6,'Nafsika');

INSERT INTO SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,
Chair,CodProgramma,NomeLocazione)
VALUES('18:40','20:00','WS-CRE-3','DenizChetinkaya@gmail.com',6,'Nafsika');

INSERT INTO SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,
Chair,CodProgramma,NomeLocazione)
VALUES('9:00','10:00','PL-4','FranciscoFlores-Revuelta@gmail.com',7,'Athena');

INSERT INTO SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,
Chair,CodProgramma,NomeLocazione)
VALUES('10:00','11:40','CSR-7','FranciscoFlores-Revuelta@gmail.com',7,'Athena');

INSERT INTO SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,
Chair,CodProgramma,NomeLocazione)
VALUES('12:00','13:20','CSR-8','FranciscoFlores-Revuelta@gmail.com',7,'Athena');

INSERT INTO SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,
Chair,CodProgramma,NomeLocazione)
VALUES('14:20','16:00','CSR-9','FranciscoFlores-Revuelta@gmail.com',7,'Athena');

INSERT INTO SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,
Chair,CodProgramma,NomeLocazione)
VALUES('10:00','11:40','WS-EPES-SPR','DenizChetinkaya@gmail.com',7,'Nafsika');

INSERT INTO SESSIONE(OrarioInizioSessione,OrarioFineSessione,TitoloSessione,
Chair,CodProgramma,NomeLocazione)
VALUES('14:20','16:00','WS-ACTI','DenizChetinkaya@gmail.com',7,'Nafsika');

--popolo la tabella INTERVENTO:
INSERT INTO
INTERVENTO(codPartecipante,OrarioInizioIntervento,OrarioFineIntervento,CodSessione)
VALUES('AlessandroGrieco@unina.it','9:30','10:00',3);

INSERT INTO
INTERVENTO(codPartecipante,OrarioInizioIntervento,OrarioFineIntervento,CodSessione)
VALUES('LorenzoBaraldi@unina.it','15:00','16:00',16);

INSERT INTO
INTERVENTO(codPartecipante,OrarioInizioIntervento,OrarioFineIntervento,CodSessione)
VALUES('CosimoDistante@unisal.it','15:00','15:30',20);

INSERT INTO
INTERVENTO(codPartecipante,OrarioInizioIntervento,OrarioFineIntervento,CodSessione)
VALUES('FrancescoBarecchia@unina.it','10:00','10:20',23);

INSERT INTO
INTERVENTO(codPartecipante,OrarioInizioIntervento,OrarioFineIntervento,CodSessione)
VALUES('AlessandroGrieco@unina.it','10:20','10:40',23);

INSERT INTO
INTERVENTO(codPartecipante,OrarioInizioIntervento,OrarioFineIntervento,CodSessione)
VALUES('ClaudioFerrari@unica.it','13:00','13:20',15);

```

```

--popolo la tabella ponte ORGANIZZARE_L:
INSERT INTO ORGANIZZARE_L(CodConferenza,emailL)
VALUES(1,'CarmenBisogni@unisa.it');

INSERT INTO ORGANIZZARE_L(CodConferenza,emailL)
VALUES(1,'FabioNarducci@unisa.it');

INSERT INTO ORGANIZZARE_L(CodConferenza,emailL)
VALUES(1,'FedericaProiettoSalanitri@unisa.it');

INSERT INTO ORGANIZZARE_L(CodConferenza,emailL)
VALUES(1,'SimonePalazzo@unica.it');

INSERT INTO ORGANIZZARE_L(CodConferenza,emailL)
VALUES(2,'CarmenBisogni@unisa.it');

INSERT INTO ORGANIZZARE_L(CodConferenza,emailL)
VALUES(2,'FabioNarducci@unisa.it');

INSERT INTO ORGANIZZARE_L(CodConferenza,emailL)
VALUES(2,'AlessandroBruno@gmail.it');

INSERT INTO ORGANIZZARE_L(CodConferenza,emailL)
VALUES(2,'PiercarloDondi@unipa.it');

INSERT INTO ORGANIZZARE_L(CodConferenza,emailL)
VALUES(2,'VincenzoLomonaco@unipa.it');


--popolo la tabella ponte PARTECIPARE:
INSERT INTO PARTECIPARE(CodSessione,emailP)
VALUES(15,'NicolaLerme@unipa.it');

INSERT INTO PARTECIPARE(CodSessione,emailP)
VALUES(15,'ClaudioFerrari@unica.it');

INSERT INTO PARTECIPARE(CodSessione,emailP)
VALUES(16,'ClaudioFerrari@unica.it');

INSERT INTO PARTECIPARE(CodSessione,emailP)
VALUES(16,'LorenzoBaraldi@unina.it');

INSERT INTO PARTECIPARE(CodSessione,emailP)
VALUES(11,'ClaudioFerrari@unica.it');

INSERT INTO PARTECIPARE(CodSessione,emailP)
VALUES(13,'LorenzoBaraldi@unina.it');

INSERT INTO PARTECIPARE(CodSessione,emailP)
VALUES(4,'GiovanniRana@gmail.com');

INSERT INTO PARTECIPARE(CodSessione,emailP)
VALUES(7,'GiovanniRana@gmail.com');

INSERT INTO PARTECIPARE(CodSessione,emailP)
VALUES(5,'CosimoDistante@unisal.it');

```

```

INSERT INTO PARTECIPARE(CodSessione,emailP)
VALUES(20,'CosimoDistante@unisal.it');

INSERT INTO PARTECIPARE(CodSessione,emailP)
VALUES(23,'FrancescoBarecchia@unina.it');

INSERT INTO PARTECIPARE(CodSessione,emailP)
VALUES(23,'AlessandroGrieco@unina.it');


--popolo la tabella ponte PUBBLICITA':
INSERT INTO pubblicita(PartitaIVA, CodConferenza, Spesa)
VALUES('00348270133', 2, 200000);

INSERT INTO pubblicita(PartitaIVA, CodConferenza, Spesa)
VALUES('04935230963', 1, 150000);

INSERT INTO pubblicita(PartitaIVA, CodConferenza, Spesa)
VALUES('11325690151', 1, 75000);

```