

# RELAZIONE PROGETTO BASE DI DATI

*Antonino Macrì – 1045245*

*Eduard Bicego – 1069882*

## **IL FAGIOLO - Ac<sup>2</sup>mms**



# Indice

1. Abstract.....	3
2. Analisi dei Requisiti.....	3
2.1 Glossario.....	4
3. Progettazione concettuale.....	5
3.1 Descrizione entità.....	5
3.2 Descrizione relazioni.....	7
3.3 Descrizione della gerarchia.....	8
3.4 Schema E-R.....	9
4. Progettazione logica.....	10
4.1 Ristrutturazione schema E-R.....	10
4.1.1 Analisi ridondanze.....	10
4.1.2 Eliminazione gerarchia.....	10
4.1.3 Partizionamento e Accorpamento di Entità e Associazioni.....	10
4.2 Traduzione verso il modello relazionale.....	11
4.2.1 Associazioni molti a molti.....	11
4.2.2 Associazioni uno a uno.....	11
4.2.3 Associazioni uno a molti.....	11
4.2.4 Entità con identificatore esterno.....	12
4.3 Schema relazionale.....	12
4.4 Descrizione testuale schema relazionale.....	13
4.5 Schema logico in forma grafica.....	15
5. Query, procedure e trigger.....	16
5.1 Query.....	16
5.2 Procedure.....	18
5.3 Trigger.....	19
6. Interfaccia Web.....	21
6.1 Sito Web.....	21
6.2 Login.....	22

# 1. Abstract

Il progetto modella la gestione degli iscritti e la loro partecipazione alla vita dell'Associazione Cattolica Campeggio Meledo Monticello Sarego (AC<sup>2</sup>MMS).

L'AC<sup>2</sup>MMS è una libera associazione che sorge per volontà dei cittadini, i quali condividono una visione cristiana della vita, integrano in comune la propria personalità, promuovono tutte quelle attività formative, turistiche, culturali, assistenziali, ricreative che ritengono utili alla costruzione di una società fondata sul pluralismo e sulla gestione sociale di tutte le istanze dell'educazione permanente con particolare attenzione nello stimolare i giovani e gli adulti alla cura dell'ambiente ed al rispetto della natura. L'associazione propone l'organizzazione di campeggi, gestione del tempo libero e attività culturali.

## 2. Analisi dei Requisiti

Si vuole creare una base di dati che gestisca gli aderenti all'associazione, la partecipazione agli eventi organizzati dalla stessa e in base al ruolo dell'aderente il suo gruppo di appartenenza (definito tappa). In particolare si vogliono conoscere i dati delle persone che almeno una volta si sono iscritte e tracciare il loro "percorso" all'interno dell'AC<sup>2</sup>MMS nei vari anni.

Nella realtà esistente gli aderenti all'associazione si suddividono in tre tipologie: genitori, animatori e animati. Essi si distinguono per il servizio svolto e il loro ruolo all'interno degli eventi organizzati dall'associazione stessa o a cui l'associazione partecipa.

Ciascuna persona è identificata dai propri dati anagrafici e può aderire all'associazione quando vuole, l'iscrizione dura per tutto l'anno solare da distinguersi dall'anno associativo che invece va dal 1 ottobre al 30 settembre dell'anno solare successivo come un tipico anno scolastico. L'anno associativo infatti rappresenta il periodo di tempo di un percorso affrontato da un gruppo di animati insieme ai propri animatori che viene identificato con il termine tappa, da qui la necessità anche di raccogliere le informazioni dell'appartenenza di ogni aderente per ogni anno.

Inoltre la base di dati deve raccogliere i dati riguardante gli eventi organizzati dall'associazione stessa e i relativi aderenti che vi partecipano. Questi dati sono una descrizione sull'evento, un programma di massima e un tema che può essere un tema già riutilizzato in eventi passati e riproposto.

## 2.1 Glossario

**Anno associativo**= inteso un anno "solastico" da settembre a circa agosto (fino all'evento del campeggio estivo).

**Ruolo**= posizione ricoperta all'interno dell'associazione

**Genitore**= un genitore nell'associazione e' inteso come un animatore anziano con i compiti di realizzazione degli eventi. Non si occupa di alcuna tappa, non anima. Non e' quindi necessaria alcuna paternita' (o maternita'). E' l'animatore stesso che chiede di divenire Genitore.

**Animato**: ragazzo che partecipa e gode di quelle attività preparate da animatori e genitori. Non è necessariamente minorenni. L'età va dalla "fine terza media" a "fine quinta superiore" da 14-15 a 19-20.

**Tappa**: gruppo di aderenti all'associazione animatori o animati, di solito gli animati appartengono alla tappa identificata con l'annata corrispondente al loro anno di nascita.

**Evento**: evento annuale nel programma dell'anno dell'associazione, diverso da istanza di evento che rappresenta il singolo evento svoltosi.

# 3. Progettazione concettuale

## 3.1 Descrizione entità

### PERSONA

Dati personali relativi a un individuo iscritto almeno una volta all'associazione.

Attributi:

- CF: char(15) PK - numero di codice identificativo della persona e chiave primaria (CF="codice fiscale")
- Nome: varchar(40) NOT NULL - nome della persona
- Cognome: varchar(40) NOT NULL - cognome della persona
- Sesso: enum('M','F') NOT NULL - sesso della persona
- DataNascita: date NOT NULL - data di nascita della persona
- LuogoNascita: varchar(40) NOT NULL - luogo di nascita della persona
- Telefono: decimal(10,0) NOT NULL - contatto telefonico
- Email: varchar(40) DEFAULT NULL - contatto e-mail
- Parrocchia: varchar(40) NOT NULL - parrocchia di appartenenza

### ADERENTE

Persone che fanno parte dell'associazione tramite iscrizione nell'anno in corso. E' generalizzazione di Genitore, Animatore ed Animato.

Attributi:

- Persona: int(11) - numero di codice identificativo della persona
- Anno: year(4) - anno associativo in cui la persona era iscritta come aderente

PK composta(Persona, Anno)

### GENITORE

La classe Genitore e' una specializzazione della classe Aderente in cui sono contenuti tutti gli aderenti investiti del ruolo di genitore all'interno dell'associazione. Non presenta attributi.

### ANIMATORE

La classe Animatore e' una specializzazione della classe Aderente in cui sono contenuti tutti gli aderenti investiti del ruolo di animatore all'interno dell'associazione. Non presenta attributi.

## ANIMATO

La classe Animato e' una specializzazione della classe Aderente in cui sono contenuti tutti gli aderenti investiti del ruolo di animato all'interno dell'associazione. Non presenta attributi.

## ISTANZA EVENTO

Rappresenta un evento svolto o da svolgere.

Attributi:

- Evento: varchar(60) DEFAULT '-' - nome evento
- DataInizio: date NOT NULL DEFAULT '0000-00-00' - data inizio evento
- DataFine: date DEFAULT NULL - data fine evento
- Luogo: varchar(40) DEFAULT '' - luogo di svolgimento dell'evento
- NPartecipanti: smallint(6) DEFAULT '0' - numero dei partecipanti confermati
- Programma: varchar(1000) DEFAULT '' - testo esplicativo dello svolgimento dell'evento
- Tema: varchar(70) DEFAULT NULL - nome tema dell'evento

PK composta(Evento, DataInizio)

## EVENTO

Eventi fissi annuali organizzati e svolti dall'associazione.

Attributi:

- Nome: varchar(60) PK - nome evento fisso
- Descrizione: varchar(1000) DEFAULT NULL - testo che descrive l'evento nelle sue generalita'
- Stagione: enum('primavera','estate','autunno','inverno') DEFAULT NULL - periodo tipico di svolgimento dell'evento

## TEMA

Tematica su cui e' stata svolta una istanza d'evento

Attributi:

- Nome: varchar(70) PK - nome del tema
- Descrizione: varchar(1000) NOT NULL DEFAULT '' - testo che descrive il tema

## **TAPPA**

Gruppo di animati suddivisi per annata con relativi animatori

Attributi:

- numRiferimento: enum('1','2','3','4','5') DEFAULT '1' - rappresenta il punto del percorso del gruppo, valore compreso tra 1 e 5.
- annoInizio: year(4) DEFAULT '0000' - anno inizio attivita' della tappa
- annoFine: year(4) DEFAULT '0000' - anno fine attivita' della tappa
- annata: year(4) DEFAULT NULL - classe di appartenenza( anno della data di nascita)

PK composta(numRiferimento, annoInizio, annoFine)

## **ASSICURAZIONE**

Rappresenta un'assicurazione valida che permetta di partecipare agli eventi

Attributi:

- Codice Tessera: varchar(20) PK - codice della tessera assicurativa

## **3.2 Descrizione relazioni**

### **Persona-Assicurazione: Assicurato**

- Molteplicità (1,1): Una persona possiede al piu' tessera assicurativa
- Totalità: totale verso Assicurazione / Parziale verso Persona

### **Persona-Aderente: Iscrizione**

- Molteplicità (N,1): Una persona puo' essere aderente per diverse volte all'interno dell'associazione, ma la sua aderenza e' univoca durante l'arco di tempo compreso in un anno solare
- Totalità: totale verso Persona / totale verso Aderente

### **Animatore-Tappa: Animazione**

- Molteplicità (N,N): Un animatore anima diverse tappe, ed una tappa puo' essere animata da diversi animatori
- Totalità: totale verso Tappa / Parziale verso Animatore

### **Animato-Tappa: Appartenenza**

- Molteplicità (N,N): L'aderente animato nel corso di un anno puo' appartenere a piu' di una tappa
- Totalità: totale verso Animato / totale verso Tappa

### **Istanza Evento-Evento: Svolgimento**

- Molteplicità (1,N): Un evento puo' svolgersi per diverse volte, ogni istanza di evento si verifica 1 ed 1 sola volta
- Totalità: totale verso Istanza Evento / Parziale verso Evento

### **Istanza Evento-Tema: Tematica**

- Molteplicità (1,N): Un'istanza evento ha al piu' un tema
- Totalità: totale verso Istanza Evento / Parziale verso Istanza Evento

### **Aderente-Istanza Evento: Partecipazione**

- Molteplicità (N,N): Un'istanza evento ha la partecipazione di almeno un aderente, e quest' ultimo puo' partecipare a diverse istanze di evento
- Totalità: Parziale verso Istanza Evento / Parziale verso Aderente

## **3.3 Descrizione della gerarchia**

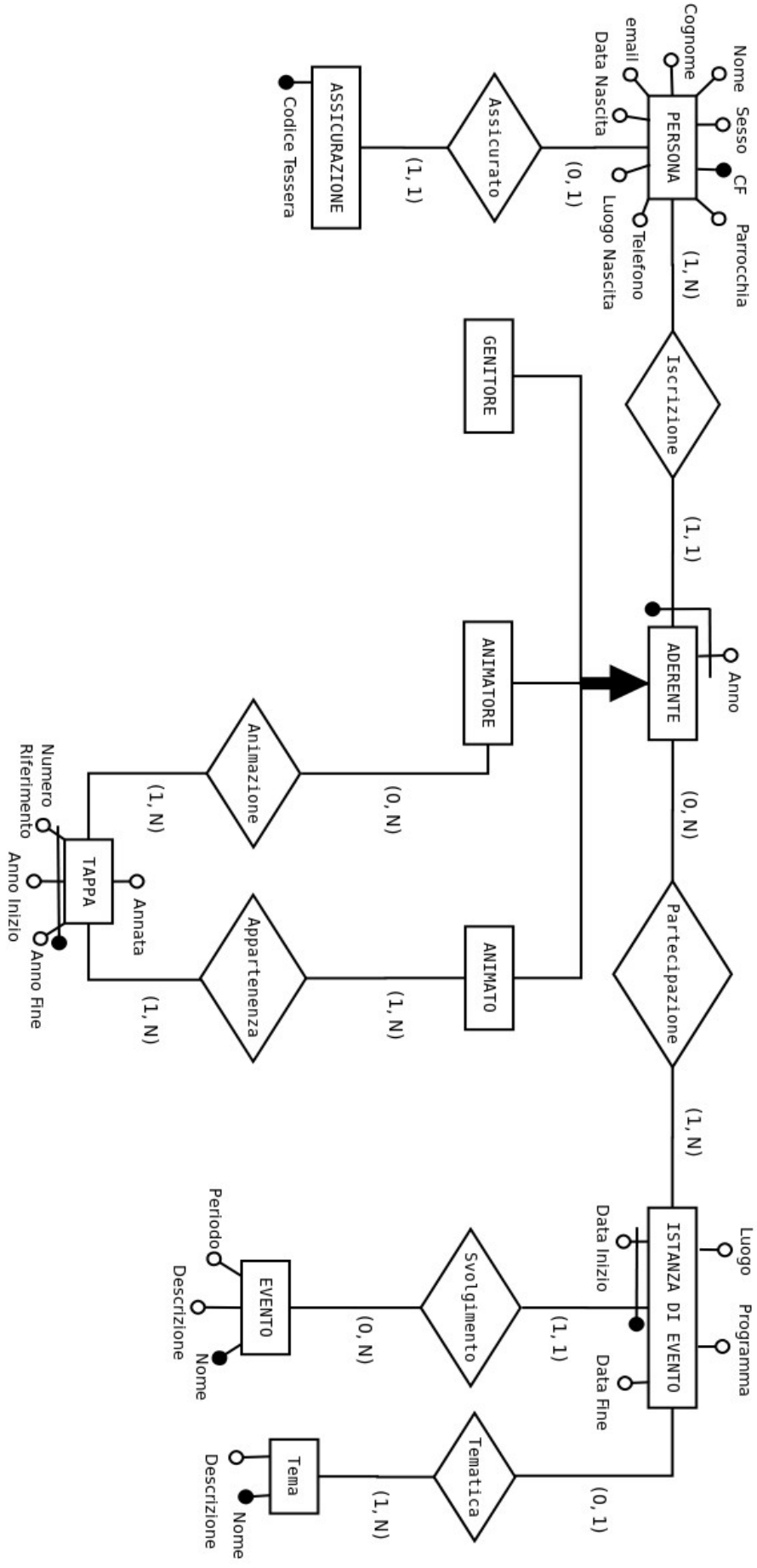
Lo schema concettuale della base di dati prevede un'unica gerarchia, quella con padre Aderente e con entita' figlie Genitore, Animatore e Animato. L'ereditarietà definisce un accorpamento in quanto le varie applicazioni soddisfano necessariamente le proprietà di:

- 1) Il padre e' sempre chiamato con le entita' figlie
- 2) La loro divisione non prevede aggiunta di attributi, infatti le loro differenze sono una caratteristica logica.

Per questo si sceglie una soluzione per accorpamento dove attraverso l'aggiunta di un attributo si puo' rappresentare il concetto delle entita' figlie pur avendo i dati dell'aderente a disposizione subito in ogni chiamata dell'aderente, senza effettuare un operazione di join che risulta essere sempre necessaria.



3.4 Schema E-R



## 4. Progettazione logica

### 4.1 Ristrutturazione schema E-R

#### 4.1.1 Analisi ridondanze

Analizzando lo schema concettuale e le possibili operazioni da fare si è deciso di introdurre un attributo che genera ridondanza. Questa decisione deriva dal fatto che l'informazione in questione risulta molto utile agli utenti per fare l'analisi dell'andamento dei partecipanti e ci sarebbe stato un elevato numero di interrogazioni per ricavarsi quel dato. Nell'entità Istanza di Evento, quindi, si introduce l'attributo Numero Partecipanti che conta il numero delle relazioni tra Aderenti e Istanza Evento.

L'attributo codice fiscale in Persona risulta essere ridondante poichè contiene i dati già presenti in altri attributi. Pertanto è stato sostituito con il campo ID che è un codice numerico con la proprietà di essere semplice.

#### 4.1.2 Eliminazione gerarchia

La gerarchia presente nel modello concettuale, avente come entità padre Aderente e come entità figlie Animatore, Animato e Genitore, è stata ristrutturata tramite accorpamento delle entità figlie nell'entità genitore a cui viene aggiunto l'attributo Ruolo per rappresentare l'informazione prima rappresentata dall'entità figlie rimosse.

L'accorpamento risulta il modo più corretto di risolvere la gerarchia perchè quest'ultima è totale ed esclusiva.

Inoltre ci assicura un minor numero di accessi in lettura e scrittura in quanto, dopo la ristrutturazione, per sapere il ruolo è sufficiente fare un accesso in lettura su Aderente, mentre prima ne servivano due per raggiungere la specializzazione a cui apparteneva l'aderente.

#### 4.1.3 Partizionamento e Accorpamento di Entità e Associazioni

Per le relazioni Animazione e Appartenenza che coinvolgono le entità Animatore e Animato accorpate in Aderente, si è deciso di accorpare anch'esse in un'unica associazione Appartenenza introducendo alcuni vincoli, questi sono:

- L'aderente con Ruolo uguale a genitore non può avere una relazione con Tappa;
- L'aderente con Ruolo uguale ad animatore può avere una relazione con

Tappa;

- L'aderente con Ruolo uguale ad animato deve avere un'appartenenza, ciò però per praticità e facilità di gestione è ignorato.

La relazione risulta quindi N a N.

Un altro accorpamento fatto è quello di Assicurazione e Assicurato collassate in un singolo attributo su Persona che dice soltanto se una persona è assicurata o meno. Questo perché il dato Codice Tessera contenuto in Assicurazione non si è ritenuto rilevante ai fini della base di dati e aumentava inutilmente la complessità dello schema.

## **4.2 Traduzione verso il modello relazionale**

### **4.2.1 Associazioni molti a molti**

Sono presenti 3 relazioni N ad N: Aderente-IstanzaEvento, Animatore-Tappa e Animato-tappa. Tutte e 3 le relazioni sono state risolte creando delle nuove entità Partecipazione ed Appartenenza che possiedono le chiavi esterne delle entità a cui sono collegate come proprie chiavi primarie.

### **4.2.2 Associazioni uno a uno**

L'unica associazione 1 ad 1 presente nel modello concettuale è persona-assicurazione. Analizzando le entità però ci si rende conto che l'importanza dell'assicurazione è semplicemente quella di garantire che un aderente abbia una assicurazione valida per partecipare alle istanze di evento. Per questa ragione l'entità è stata eliminata e sostituita da un campo in più all'interno dell'entità aderente che rispondesse alla domanda di poter partecipare con un 'si' 'no'.

### **4.2.3 Associazioni uno a molti**

Le 3 associazioni di tipo 1 ad N sono: Persona-Aderente, IstanzaEvento-Evento ed IstanzaEvento-Tema. Tutte e tre queste associazioni non necessitano di campi aggiuntivi in più.

La traduzione delle associazioni 1 a molti prevede che agli attributi della relazione che partecipa con cardinalità massima pari a uno venga aggiunto l'identificatore della relazione che partecipa con cardinalità massima pari a N (ed eventuali attributi dell'associazione stessa, ma non è il nostro caso). Quindi la risoluzione è stata proprio quella descritta inserendo in IstanzaEvento le chiavi esterne Tema(nome <PK>) ed Evento(nome <PK>) ed in Aderente la chiave esterna di Persona(ID <PK>).

## 4.2.4 Entità con identificatore esterno

Essendo Aderente un'entità con identificatore esterno verrà tradotta nel modello relazione come una relazione con chiave che include gli identificatori delle entità "identificanti", cioè Persona; in questo modo verrà rappresentata sia l'entità coinvolta e anche l'associazione Iscrizione.

Gli identificatori che verranno inclusi nella relazione Aderente andranno a comporre la chiave primaria della relazione insieme all'identificatore interno anno.

Lo stesso discorso si può fare per l'entità IstanzaEvento che prende come chiave primaria oltre all'identificatore interno annoInizio anche l'identificatore esterno nome di Evento.

## 4.3 Schema relazionale

- ◆ PERSONA(id, nome, cognome, sesso, dataNascita, luogoNascita, telefono, email, parrocchia, assicurato)
- ◆ ADERENTE(persona, anno, ruolo)
- ◆ TAPPA(numRiferimento, annoInizio, annoFine, annata)
- ◆ APPARTENENZA(aderentePersona, aderenteAnno, tappaNumRiferimento, tappaAnnoInizio, tappaAnnoFine)
- ◆ PARTECIPAZIONE(persona, anno, evento, dataInizio)
- ◆ ISTANZAEVENTO(evento, dataInizio, dataFine, nPartecipanti, luogo, programma, tema)
- ◆ EVENTO(nome, descrizione, stagione)
- ◆ TEMA(nome, descrizione)

## 4.4 Descrizione testuale schema relazionale

PERSONA (id: int(11), nome: varchar(40), cognome varchar(40), sesso enum('M','F'), dataNascita: date, luogoNascita varchar(40), telefono decimal(10,0), email varchar(40), parrocchia varchar(40), assicurato: enum('si','no'))

- Id=AUTO\_INCREMENT
- PK (id)
- NOT NULL Nome, Cognome, Sesso, dataNascita, luogoNascita, telefono, email, parrocchia

ADERENTE (persona: int(11), anno: year(4), ruolo: enum('AR','AN','GE'))

- PK (persona, anno)
- persona FK persona(id)
- NOT NULL Ruolo

TAPPA (numRiferimento: enum('1','2','3','4','5'), annoInizio: year(4), annoFine: year(4), annata: year(4))

- PK (numRiferimento, annoInizio, annoFine)
- NOT NULL Annata

APPARTENENZA (aderentePersona: int(11), aderenteAnno: year(4), tappaNumRiferimento: enum('1','2','3','4','5'), tappaAnnoInizio: year(4), tappaAnnoFine: year(4))

- PK (aderentePersona, aderenteAnno, annoInizio, annoFine)
- annoInizio, annoFine, tappaNumRiferimento FK tappa(tappaAnnoInizio, tappaAnnoFine, numRiferimento)
- aderentePersona, aderenteAnno FK aderente(persona, anno)

PARTECIPAZIONE (persona: int(11), anno: year(4), evento: varchar(60), dataInizio: date)

- PK (aderente, anno, evento, dataInizio)
- persona, anno FK aderente(persona, anno)
- evento, dataInizio FK istanzaevento(nome, dataInizio)

ISTANZAEVENTO (evento: varchar(60), dataInizio: date, dataFine: date, nPartecipanti: smallint(6), luogo: varchar(40), programma: varchar(1000), tema: varchar(70))

- PK (evento, dataInizio)
- evento FK evento(nome)
- tema FK tema(nome)
- nPartecipanti >= 0

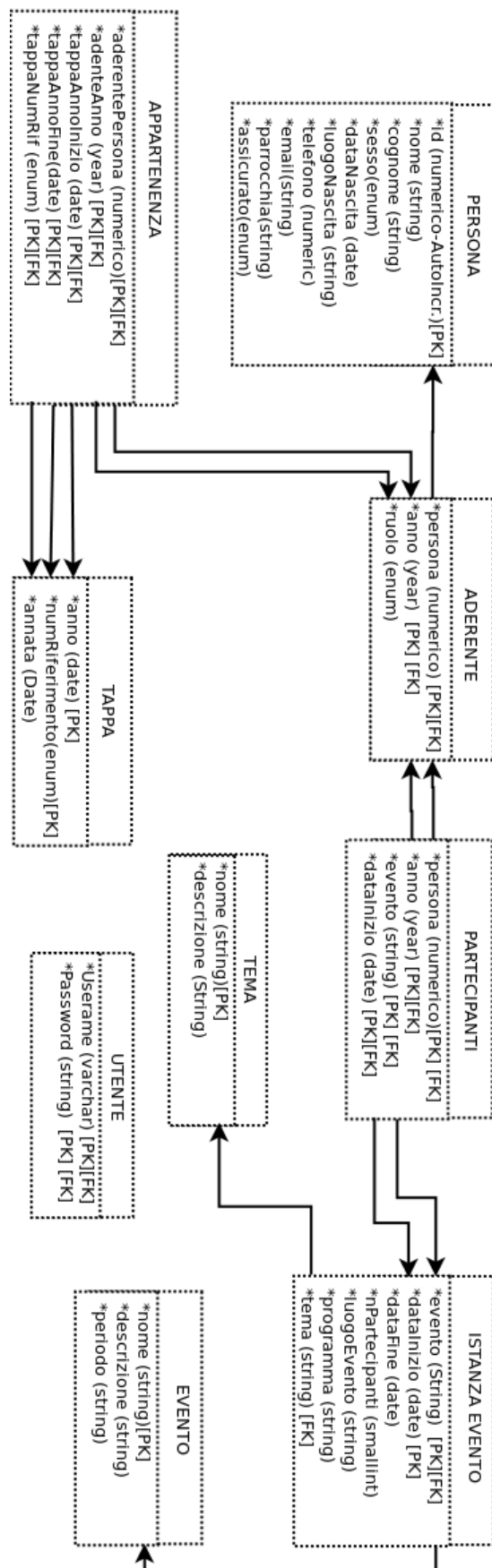
EVENTO (nome: varchar(60), descrizione: varchar(1000), stagione:  
enum('primavera','estate','autunno','inverno'))

- PK (nome)

TEMA(nome: varchar(70), descrizione: varchar(1000))

- PK (nome)
- NOT NULL descrizione

## 4.5 Schema logico in forma grafica



# 5. Query, procedure e trigger

## 5.1 Query

1. Trovare l'istanza dell'evento svolto piu' recentemente alla data odierna e l'istanza dell'evento piu' prossimo alla data odierna.

```
SELECT evento, dataInizio, luogo
FROM IstanzaEvento
WHERE dataInizio=(SELECT MIN(dataInizio) AS dataInizio
FROM IstanzaEvento
WHERE dataInizio>CURDATE())
UNION
SELECT evento, dataInizio, luogo
FROM IstanzaEvento
WHERE dataInizio=(SELECT MAX(dataInizio) AS dataInizio
FROM IstanzaEvento
WHERE dataInizio<CURDATE());
```

2. L'aderente CURDATE() che ha partecipato il maggior numero di volte agli eventi CURDATE():

```
SELECT MAX(F.numPartecipazioni) AS maxPartecipazioni, E.id, E.nome, E.cognome,
E.dataNascita
FROM (SELECT COUNT(P.persona) AS numPartecipazioni, P.persona
FROM partecipazione AS P
WHERE P.anno=YEAR(CURDATE()) && YEAR(P.dataInizio)=YEAR(CURDATE())
GROUP BY P.persona) AS F JOIN persona AS E ON F.persona=E.id
```

3. L'id, il nome, il cognome e la data di nascita di tutti gli aderenti dell'anno in corso che siano assicurati e che non partecipino all'evento specificato con '\$evento' e '\$dataInizio':

```
SELECT id,nome,cognome,dataNascita
FROM persona JOIN aderente ON Id=Persona
WHERE assicurato='si' AND anno=YEAR(CURDATE())
AND id NOT IN (SELECT PE.id
FROM persona AS PE JOIN aderente AS A ON PE.id=A.persona
```



```

JOIN partecipazione AS P ON A.persona=P.persona
AND A.anno=P.anno JOIN istanzaevento AS I ON
      P.dataInizio=I.dataInizio AND P.evento=I.evento
WHERE I.evento='$evento' AND I.dataInizio='$dataInizio')

```

4. Il numero dei componenti delle tappe con ruolo di animato, affiancato dall'anno di inizio e di fine e il numero di riferimento della tappa corrispondente ordinato in modo decrescente rispetto l'anno inizio e l'anno fine.

```

SELECT count(*) AS numeroAN, T.annoInizio, T.annoFine, T.numRiferimento
FROM aderente AS A JOIN persona AS P ON A.persona=P.id JOIN appartenenza AS AP
ON   AP.aderentePersona=A.persona AND AP.aderenteAnno=A.anno JOIN tappa AS T
ON   T.annoInizio=AP.tappaAnnoInizio AND T.annoFine=AP.tappaAnnoFine
      AND T.numRiferimento=AP.tappaNumRif
WHERE A.ruolo='AN'
GROUP BY T.annoInizio, T.annoFine, T.numRiferimento
ORDER BY T.annoInizio DESC, T.annoFine DESC

```

5. L'evento e la stagione di ogni evento annuale con la media arrotondata per eccesso dei partecipanti.

```

SELECT I.evento, E.stagione, CEIL(AVG(I.nPartecipanti)) mediaPartecipanti
FROM evento AS E JOIN istanzaevento AS I ON E.nome=I.evento
GROUP BY I.evento

```

6. Il nome dell'evento e la data di inizio delle istanze evento con l'affluenza calcolata in percentuale ordinate in modo decrescente in base alla data di inizio.

```

SELECT evento, dataInizio, nPartecipanti/nAderenti*100 AS affluenza
FROM (SELECT evento, nPartecipanti, dataInizio, YEAR(dataInizio) AS data
      FROM istanzaevento ) AS I JOIN
      (SELECT COUNT(anno) AS nAderenti, anno
      FROM aderente
      GROUP BY anno) AS A ON I.data=A.anno
ORDER BY anno DESC

```

## 5.2 Procedure

1. Crea automaticamente, tramite input di un data rappresentabile l'annata, le 5 tuple che sono il percorso dell'annata nelle 5 tappa dalla durata di un anno scolastico ognuna (5 anni).

```
|CREATE PROCEDURE inserisci_annata(IN newannata int(4))
BEGIN
    INSERT INTO `tappa` (`numRiferimento`, `annoinizio`, `annofine`, `annata`)
    VALUES ('1', newannata+14, newannata+15, newannata),
    ('2', newannata+15, newannata+16, newannata),
    ('3', newannata+16, newannata+17, newannata),
    ('4', newannata+17, newannata+18, newannata),
    ('5', newannata+18, newannata+19, newannata);
END |
```

2. Imposta il campo assicurato di tutte le persone a 'no'.

```
|CREATE PROCEDURE insurance_flag(IN flag char(2))
BEGIN
    update persona
    set assicurato=flag;
END |
```

3. Controllo che l'aderente passato in input abbia campo assicurato='si' per poterlo inserire in partecipazione altrimenti la partecipazione è rifiutata.

```
|CREATE PROCEDURE flag_assicurato
    (IN par_id int(11),IN par_evento varchar(40), OUT inserito int(1))
BEGIN
    DECLARE dataevento char(10);
    DECLARE flag char(2);
    select assicurato into flag
    from persona where id=par_id;
    select dataInizio into dataevento
    from istanzaevento
    where evento=par_evento AND dataInizio>curdate();
    if(flag='si') then
        insert into partecipazione values
        (par_id,year(curdate()),dataevento,par_evento);
        set inserito=1;
    else
        set inserito=0;
    end if;
END |
```

## 5.3 Trigger

1. Ogni partecipante aggiunto a una istanza evento in partecipazione aziona il trigger che aumenta di 1 il valore dell'attributo nPartecipanti dell'istanza evento presa in causa.

```
|  
CREATE TRIGGER conta_partecipanti  
AFTER INSERT ON partecipazione  
FOR EACH ROW  
BEGIN  
    UPDATE istanzaevento  
    SET nPartecipanti=nPartecipanti+1  
    WHERE evento=new.evento && dataInizio>curdate();  
END; |
```

2. Ogni partecipante eliminato da una istanza evento in partecipazione aziona il trigger che decrementa di 1 il valore dell'attributo nPartecipante dell'istanza evento presa in causa.

```
|  
CREATE TRIGGER togl_partecipanti  
BEFORE DELETE ON partecipazione  
FOR EACH ROW  
BEGIN  
    DECLARE VAR INT;  
    SELECT nPartecipanti from istanzaevento  
    WHERE evento=old.evento && dataInizio>curdate() into var;  
    if(var>0) then  
        UPDATE istanzaevento  
        SET nPartecipanti=nPartecipanti-1  
        WHERE evento=old.evento && dataInizio>curdate();  
    end if;  
END; |
```

3. Se viene inserito in appartenenza un utente già appartenente ad una tappa, viene cancellata la sua appartenenza alla precedente tappa prima di inserire l'utente. In caso di inserimento corretto allora esegua la normale operazione di incrementare il numero di partecipanti

```
|  
CREATE TRIGGER `conta_partecipanti`  
AFTER INSERT ON `partecipazione`  
FOR EACH ROW begin  
    DECLARE var VARCHAR(2);  
    SELECT pe.assicurato into var  
    FROM aderente a join persona pe on pe.id=a.persona  
    WHERE a.persona=new.persona and a.anno=new.anno;  
    if(var='no') then  
        DELETE FROM partecipazione  
        WHERE persona=new.persona and anno=new.anno;  
    else  
        UPDATE istanzaevento  
        SET nPartecipanti=nPartecipanti+1  
        WHERE evento=new.evento && dataInizio>curdate();  
    end if;  
END; |
```

# 6. Interfaccia Web

La progettazione dell'interfaccia web si è concentrata in tre aspetti che rappresentano le tre tipologie di utenti che possono visitare il sito e accedere quindi alle informazioni contenute nel database. Questi sono:

- **area pubblica** che è la parte del sito in cui avvengono interrogazioni basilari e limitate al database.
- **area riservata** che a sua volta si divide in area riservata soci e area riservata amministrazione, nella prima troviamo i servizi che il database offre a chi è socio dell'associazione e sono solo di lettura mentre nella seconda troviamo i servizi di scrittura per aggiornare il database con nuovi dati o con la rimozione di essi.

Per l'interfaccia web si sono utilizzati gli standard CSS, per rendere gradevole la vista del sito, e XHTML con alcuni script Javascript utilizzati per migliorare l'esperienza dell'utente.

## 6.1 Sito Web

Le pagine del sito web sono strutturate in modo da renderle semplici e migliorabili facilmente nel futuro. Il sito è strutturato da pagine php costruite tramite funzioni raccolte in file print.php per evitare qualsiasi duplicazione del codice e sfruttare funzioni già scritte, se possibile. Infatti tutto ciò ha reso molto semplice l'inserimento di nuove pagine e la modifica di parti piccole e grandi di ogni pagina.

Lo scheletro del sito è il seguente:

- **Area pubblica:** contiene il front end del sito accessibile a qualunque tipo di utente che si collega al sito. Essa rende disponibile l'accesso solo ad alcuni dati del database quali: il prossimo evento programmato e l'ultimo avvenuto, gli animatori con un contatto delle rispettive tappe dell'anno e gli eventi con (quasi) cadenza annuale organizzati dall'associazione.
- **Area riservata**, divisa a sua volta in:
  - **Area socio:** rende disponibili le interrogazioni basilari e avanzate alla base di dati per avere un resoconto dell'andamento dell'associazione per aderenti e partecipanti. Non è permessa alcun tipo di modifica.
  - **Area admin:** dispone di tutti i servizi per agevolare l'amministrazione della base di dati, qui è infatti possibile aggiungere nuovi dati, modificarli e eliminarli tramite un'interfaccia intuitiva.

## 6.2 Login

Le credenziali d'accesso per le aree riservate del sito sono:

- Per AREA SOCIO
  - Username: admin
  - Password: admin
  
- Per AREA ADMIN
  - Username: socio
  - Password: socio