

Esame di stato

ARTICOLAZIONE INFORMATICA

Analisi:

Viene richiesta la realizzazione di una web community con lo scopo di condividere dati e commenti relativi ad eventi dal vivo di vario genere, come concerti o spettacoli teatrali, con ubicazione in Italia.

L'inserimento degli eventi viene effettuato direttamente dai membri iscritti della community, la cui registrazione avviene tramite l'inserimento di dati anagrafici, un nickname e una o più categorie di eventi a cui sono interessati.

In seguito all'iscrizione si riceverà periodicamente una newsletter, ricevuta tramite posta elettronica. Tale newsletter viene inviata automaticamente dal sistema, contenente gli eventi delle categorie scelte che si svolgeranno nel territorio dell'utente nella settimana seguente.

I membri che hanno deciso di registrarsi alla community hanno la possibilità di interagire inserendo nuovi eventi, per il quale occorre specificare categoria, luogo di svolgimento, data, titolo dell'evento e artisti coinvolti, oppure scrivere un post con un commento ed un voto (da 1 a 5) su un evento.

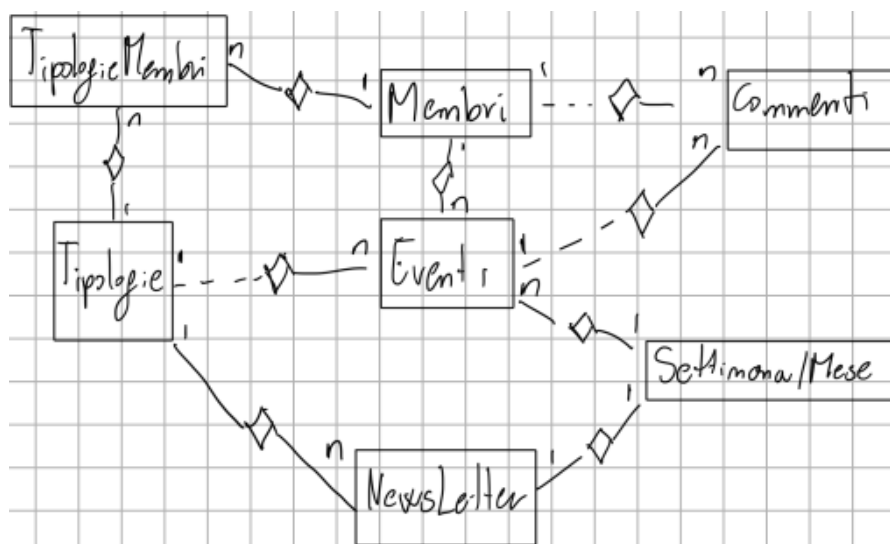
La community rende accessibile sia ai membri registrati che a quelli anonimi, la consultazione di dati on line tra cui la visualizzazione degli eventi di un certo tipo in ordine cronologico, con possibilità di filtro per territorio di una specifica provincia e la visualizzazione di tutti i commenti e voti relativi ad un evento.

Si vede necessario implementare un'interfaccia web per la consultazione degli eventi all'interno della community e le operazioni possibili dai vari tipi di utente. Inoltre bisogna implementare la multiutenza di accesso al database data la possibilità di avere più utenti che ci lavorano contemporaneamente.

Ipotesi aggiuntive:

Può essere inserito un database per la gestione dei membri che si registrano, con nickname e password. In questo modo le informazioni relative alle credenziali possono essere criptate con un algoritmo di HASH.

La gestione della newsletter viene gestita dal database tramite un'entità. Può essere tutta via affidata a programmi di terzi che inviino email in automatico ogni mese con un corpo uguale per tutti.

Schema concettuale:

Regole di lettura:

Dato un membro questo può inserire zero o più eventi,
dati tanti eventi, questi possono essere scritti da un solo membro
Dato un commento, questo può essere scritto da un solo membro,
dato un membro, questo può scrivere zero o tanti commenti.
Dato un evento, questo può avere zero o tanti commenti,
dato un commento, questo può essere scritto per un solo evento.
Data una settimana/mese, questa può essere associata a più eventi,
dato un evento, questo può essere associato ad una sola settimana/mese.
Data una newsletter, questa può essere associata ad una sola settimana/mese,
data una settimana/mese, questa ha una sola newsletter associata.
Data una newsletter, questa ha una sola tipologia,
data una tipologia, questa può essere assegnata a più newsletter.
Data una tipologia, questa può essere associata a molti eventi,
dato un evento questo può avere zero o una tipologia.
Data una tipologia, abbiamo tante tipologia/membri,
data una tipologia/membri, abbiamo una tipologia.
Dato un membro abbiamo tante tipologia/membri,
data una tipologia/membri, abbiamo un membro.

Schema logico:

tblCategorie (idTipologia, descrizione);
tblMembri (nickname, password, cognome, nome, email, provincia);
tblTipologieMembri (nickname*, idTipologia*);
tblNewsLetter (nickname*, idTipologia*, idEvento*, data*, descrizione);
tblEventi (idEvento, nickname*, idTipologia*, titolo, luogo, dataInizio*, dataFine*, artista, descrizione);
tblCommenti (idEvento*, nickname*, commento, voto, locandina);
tblSettimana/mese (settimana, mese)

Creazione tabelle:

Create table tblCategorie(

idTipologia integer primary key auto_increment,
descrizione varchar (300) not null
);

Create table tblMembri(

Nickname varchar (10) not null,
Password varchar (10),
Cognome varchar (10),
Nome varchar (10),
Email varchar (30) not null,
Provincia varchar (20) not null
);

Create table tblTipologieMembri(

Nickname varchar (10) references tblMembri (nickname),
idTipologia integer references tblCategorie (idTipologia),
foreign key (nickname) references tblMembri (nickname),
foreign key (idTipologia) references tblCategorie (idTipologia)
primary key (nickname, idTipologia)
);

Create table tblEventi(

idEvento integer primary key auto_increment,
Nickname varchar (10) references tblMembri (nickname),
idTipologia integer references tblCategorie (idTipologia),
titolo varchar (30) not null,
luogo varchar (50) not null,
dataInizio date not null,
dataFine date not null,
artista varchar (50),
descrizione varchar (200),
foreign key (nickname) references tblMembri (nickname),
foreign key (idTipologia) references tblCategorie (idTipologia),

```
foreign key (dataInizio) references tblSettimanaMese (settimana,mese),
foreign key (dataFine) references tblSettimanaMese (settimana,mese),
primary key (idEvento, Nickname, idTipologia)
);
```

Create table tblNewsLetter(

```
Nickname varchar (10) references tblMembri (nickname),
idTipologia integer references tblCategorie (idTipologia),
idEvento integer references tblEventi(idEvento),
data date not null,
foreign key (data) references tblSettimanaMese (settimana, mese),
foreign key (nickname) references tblMembri (nickname),
foreign key (idTipologia) references tblCategorie (idTipologia),
foreign key (idEvento) references tblEventi (idEvento)
);
```

create table tblCommenti(

```
idEvento integer references tblEvento (idEvento),
Nickname varchar (10) references tblMembri (nickname),
commento varchar (200) not null,
voto integer not null,
locandina blob
```

);

Create table tblSettimanaMese(

```
settimana integer not null,
mese date not null,
primary key (settimana,mese)
);
```

Query:

1

```
Select *  
From tblEventi, tblMembri  
Where tblEventi.nickname = tblMembri.nickname  
And data < curdate ()  
ORDER BY provincial;
```

2

```
Select *  
From tblMembri  
Where nickname not in (  
                Select distinct nickname  
                From tblCommenti  
                );
```

3

```
Select eventi.idEvento, titolo, avg (voto) as media  
From tblCommenti, tblEventi  
Where tblEventi.idEvento = tblCommenti.idEvento  
Group by tblEventi.idEvento, titolo  
Order by idTipologia, titolo;
```

4

Select nickname

From (

 Select nickname, count (*) as nEventi

 From tblEventi

 Group by nickname

) as n1

Where nEventi in (

 Select max (nEventi)

 From (

 Select nickname, count (*) as nEventi

 From tblEventi

 Group by nickname

) as n2

);