

Esame di stato 2014-2015

Indirizzo: ITIA-INFORMATICA E TELECOMUNICAZIONI

ARTICOLAZIONE INFORMATICA

Tema di: INFORMATICA

SOLUZIONE 1 PARTE

IPOTESI AGGIUNTIVE

Essendo lo sviluppo completo della traccia a livello professionale non realizzabile in 6 ore si fanno le seguenti assunzioni semplificative:

- Per i database si utilizzeranno sole le informazioni strettamente necessarie
- Per il codice (script) si riportano sole le istruzioni essenziali non tenendo conto di problemi di sicurezza, gestione dei controlli sui form, omogeneizzazione dei dati.

1-ANALISI DELLA REALTA' DI RIFERIMENTO, POSSIBILI SOLUZIONI, MOTIVAZIONE DELLA SOLUZIONE SCELTA

Si possono prendere come riferimento le varie community che si sono succedute negli anni come: myspace, mymusic o le pagine Facebook.

Come modello di "informazione" circa gli eventi si possono prendere le pagine del comune di Torino sui concerti o di TiketOne o di Torinosette. Quest'ultima forse è la più simile al modello proposto dal testo.

Il tipo di approccio è senza dubbio una web application che può essere sviluppata mediante un CMS (Wordpress, Drupal, Joomla) integrato con plugin personalizzati o mediante un progetto ex-novo

Per semplicità si opta per la seconda soluzione.

Si adottano, al fine di contenere i costi, le seguenti soluzioni:

Web hosting: Altvista.org (gratuito) che offre la seguenti tecnologie:

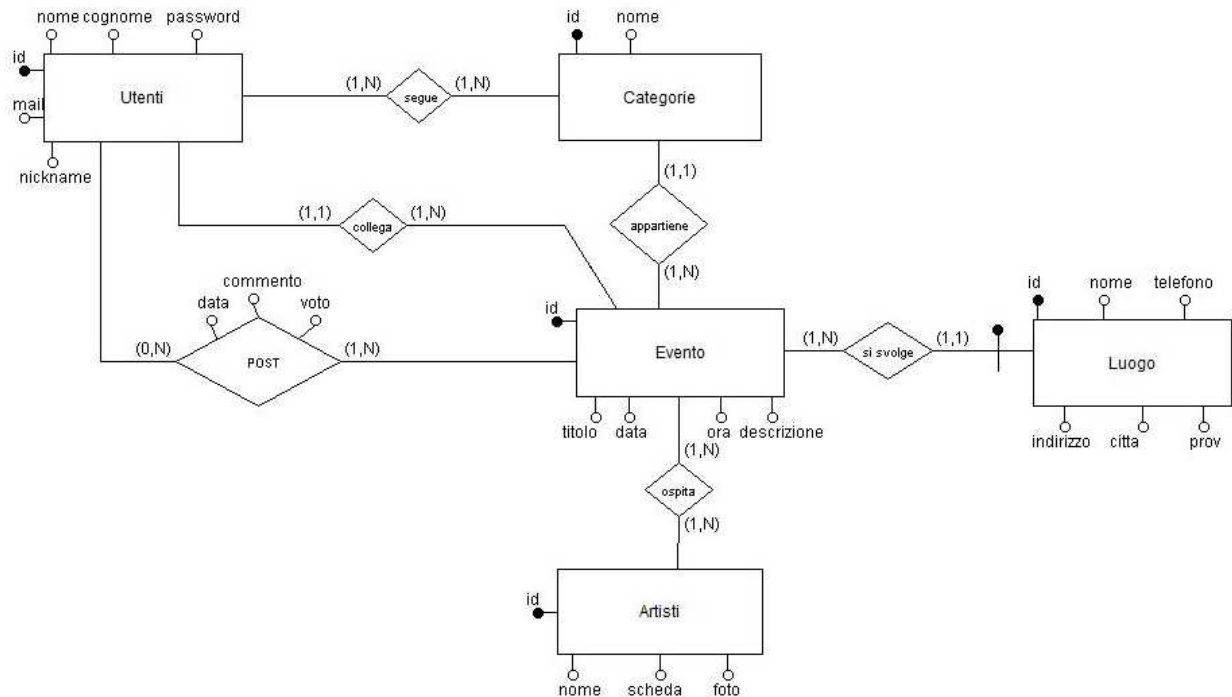
- Database MySQL e phpMyAdmin per la gestione dello stesso
- Motore PHP
- Editor on line per gli script
- Accesso via FTP (per sviluppo in locale)
- Gestione banner pubblicitari

Ricerca su web di un Template gratuito di tipo responsive con intestazione, struttura a due/tre colonne nel corpo centrale e pie' pagina

Sviluppo in locale:

- XAMPP per simulare ambiente di sviluppo compatibile con Altrivista.org (web server Apache, motore PHP, database MySQL, phpMyAdmin)
- Dreamviewer (scrittura script HTML, CSS, JS, PHP, JQUERY) con sincronizzazione via FTP con host su Altrivista
- FileZilla per connessione FTP con host su Altrivista

2-SCHEMA CONCETTUALE DELLA BASE DI DATI



Relazioni:

SEGUE: Un utente può scegliere più categorie. Ogni categoria può essere abbinata a più utenti. Rel. N-N

POST: Un utente scrive 0 o più commenti ad uno o più eventi. Un evento può avere uno o più commenti/voti. Rel. N-N

COLLEGA: Un utente può inserire 1 o più eventi. Un evento appartiene ad un singolo utente. Rel. 1-N

OSPITA: Un evento può ospitare diversi artisti. Un artista può partecipare a più eventi. Rel. N-M

SI SVOLGE: Nello stesso luogo si possono svolgere più eventi (in date diverse). Un dato evento si svolge in un unico luogo. Rel. 1-N

APPARTIENE: Un dato evento appartiene ad una sola categoria. Ci possono essere più eventi diversi appartenenti alla medesima categoria. Rel. 1-N

3-SCHEMA LOGICO

UTENTI (id, nome, cognome, nickname, password, email) –Contiene gli utenti registrati

UTENTI_CATEGORIE(id_utente,id_categoria) – Associazione N-N tra le due entità. Vedi nota (1)

CATEGORIE(id, nome) – Elenco delle categorie (Musica, Teatro, Balletto, ecc.)

POST (data, commento, voto, id_utenteFK, id_eventoFK) – Associazione dei commenti agli eventi

EVENTI(id, titolo, data, ora, descrizione, id_categoriaFK, id_utenteFK, id_luogoFK)-Memorizza evento

LUOGO (id, nome, indirizzo, citta', prov, telefono) – Luogo in cui si svolge l'evento

ARTISTI (id, nome, scheda, foto) – Elenco degli artisti che partecipano agli eventi

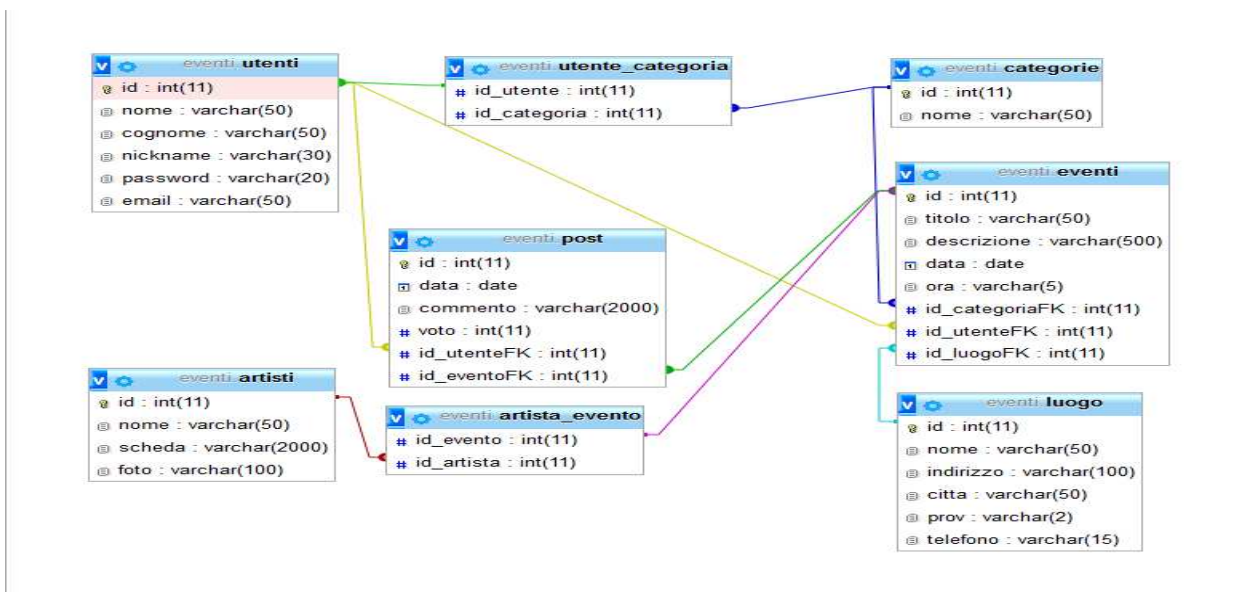
ARTISTA_EVENTO (id_evento, id_artista) – Associazione N-N tra le due entità . vedi nota (2)

Note:

1. Traduce la relazione N-N tra le entità UTENTI e CATEGORIE. Rappresenta il rombo "Segue"

2. Traduce l'associazione N-N tra le entità EVENTI e ARTISTI. Rappresenta il rombo "Ospita"

Schema con relazioni (da phpMyAdmin)



4-DEFINIZIONE IN SQL DI UN SOTTOINSIEME DELLE RELAZIONI DELLA BASE DI DATI CON VINCOLI DI INTEGRITA' REFERENZIALE E/O DI DOMINIO

Si riportano a titolo di esempio lo sviluppo di tre tabelle:

TABELLA UTENTI	TABELLA POST	TABELLA EVENTI
<pre>CREATE TABLE utenti (id int(11) NOT NULL AUTO_INCREMENT, nome varchar(50) NOT NULL, cognome varchar(50) NOT NULL, nickname varchar(30) NOT NULL, `password` varchar(20) NOT NULL, email varchar(50) NOT NULL, PRIMARY KEY (id)) ENGINE=InnoDB;</pre>	<pre>CREATE TABLE post (id int(11) NOT NULL AUTO_INCREMENT, `data` date NOT NULL, commento varchar(2000) NOT NULL, voto int(11) NOT NULL, id_utenteFK int(11) NOT NULL, id_eventoFK int(11) NOT NULL, PRIMARY KEY (id), KEY id (id), KEY id_utenteFK (id_utenteFK), KEY id_eventoFK (id_eventoFK)) ENGINE=InnoDB</pre>	<pre>CREATE TABLE eventi (id int(11) NOT NULL AUTO_INCREMENT, titolo varchar(50) NOT NULL, descrizione varchar(500) NOT NULL, `data` date NOT NULL, ora varchar(5) NOT NULL, id_categoriaFK int(11) NOT NULL, id_utenteFK int(11) NOT NULL, id_luogoFK int(11) NOT NULL, PRIMARY KEY (id), KEY id (id), KEY id_categoriaFK (id_categoriaFK), KEY id_utenteFK (id_utenteFK), KEY id_luogoFK (id_luogoFK)) ENGINE=InnoDB</pre>

RELAZIONE ED INTEGRITA REFERENZIALE TRA LE TABELLE RIFERITA ALLA TABELLA POST

```
ALTER TABLE post
ADD CONSTRAINT post_ibfk_2 FOREIGN KEY (id_eventoFK) REFERENCES `eventi` (id) ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT post_ibfk_1 FOREIGN KEY (id_utenteFK) REFERENCES utenti (id) ON DELETE CASCADE ON UPDATE CASCADE;
```

NOTA: In MySQL per realizzare le relazioni è necessario che tutte le chiavi esterne siano indicizzate (KEY...) e che tutte le tabelle siano di tipo InnoDB

5-INTERROGAZIONI IN LINGUAGGIO SQL

- a) Elenco degli eventi già svolti, in ordine alfabetico di provincia.

Nell'ipotesi che la query venga fatta il giorno: 18 giugno 2015

```
SELECT titolo, data, prov
FROM eventi, luogo
WHERE eventi.`id_luogoFK` = luogo.id
AND eventi.`data` < "2015-06-18"
ORDER BY prov
```

- b) Elenco dei membri che non hanno mai inserito un commento

```
SELECT nome, cognome
FROM utenti, post
WHERE post.id_utenteFK <> utenti.id
```

- c) Per ogni evento il voto medio ottenuto in ordine di categoria e titolo

```
SELECT titolo,nome as "categoria", avg(post.voto) as "Voto medio" FROM
eventi, post, categorie
WHERE post.id_eventoFK=eventi.id
AND eventi.id_categoriaFK=categorie.id
GROUP BY eventi.id
ORDER BY categorie.nome, eventi.titolo
```

d) I dati dell'utente che ha registrato il maggior numero di eventi

Questa query si compone di due parti:

1) si crea una vista con il conteggio degli eventi registrati da ogni utente

```
CREATE VIEW eventi_x_utente AS SELECT nome, cognome, email, count(
id_utenteFK ) AS "num_eventi"
FROM eventi, utenti
WHERE utenti.id = eventi.id_utenteFK
GROUP BY id_utenteFK
```

2) si estrae il massimo dalla vista

```
SELECT nome, cognome, email, max( num_eventi ) AS "tot_eventi"
FROM eventi_x_utente
```

si può poi cancellare la vista con : DROP VIEW eventi_x_utente

OPPURE, MOLTO PIU' SEMPLICEMENTE:

```
SELECT nome, cognome, email, count( id_utenteFK ) AS "num_eventi"
FROM eventi, utenti
WHERE utenti.id = eventi.id_utenteFK
GROUP BY id_utenteFK
ORDER BY num_eventi DESC
LIMIT 1
```

6-PROGETTO DELL'INTERFACCIA WEB CHE PERMETTA ALL'UTENTE REGISTRATO DI SVOLGERE OPERAZIONI SPECIFICHE

LOGO E NOME DEL SITO		
BARRA MENU: HOME-LOGIN/LOGOUT,-REGISTRATI-RICERCA EVENTO(2)-CONTATTI-.....		
MENU UTENTE (1) Mario Rossi(Esci) Profilo I miei eventi I mie artisti I miei luoghi Inser. Luogo Inser. Artista Inser. Evento	PAGINA INSERIMENTO EVENTO (3) Titolo evento: [casella di testo] – Data evento: [casella con calendario] Ora inizio: [casella di testo] – Categoria: [select](a) Luogo: [select](b) – Descrizione: [text area] Artisti: ora inizio[text] – Artista: [select](c) ora inizio[text] – Artista: [select](c) ora inizio[text] – Artista: [select](c)(max 10 artisti) [INVIA] [CANCELLA] (d)	BANNER NEWS ALTRO

Essendo il codice lungo e complesso non è possibile in questa sede darne un esempio

NOTE:

- 1) Menu utente: compare dopo il login.
 - a. Profilo: permette la modifica di email, pw o altri dati dell'utente
 - b. I miei eventi: elenco degli eventi inseriti dall'utente con possibilità (mediante 3 link) di dettaglio, modifica dati o cancellazione
 - c. I mie artisti: come punto b ma per gli artisti. La cancellazione è possibile solo se l'artista non è inserito in alcun evento
 - d. I miei luoghi: come punto c ma per i luoghi dove si svolge l'evento. No cancellazione luogo se utilizzato in un evento

- e. Inserimento Luogo: form per inserimento dati del luogo (nome, indirizzo, città, ecc)
 - f. Inserimento Artista: form ins. dati dell'artista (nome, scheda, foto)
 - g. Inserimento evento: form dati evento (titolo, descrizione, data, ora, ecc)
 - h. Il logout si puo fare mediante il link "Esci" o dalla barra di navigazione (LOGOUT)
- 2) Inserimento POST mediante RICERCA EVENTI: Utilizzando la normale ricerca eventi disponibile sulla barra dei menù e visualizzando il dettaglio dell'evento, se l'utente è autenticato, sotto il dettaglio comparirà la possibilità di inserire un commento e dare il voto. Se invece l'evento visualizzato dalla ricerca appartiene all'utente al posto del commento/voto ci saranno i link per la modifica/cancellazione. La modifica o cancellazione dell'evento è possibile solo se l'evento deve ancora svolgersi
- 3) Esempio di form per la registrazione evento. Prima di registrare l'evento l'utente deve inserire il luogo e gli artisti con le apposite voci di menu. Tutti i campi sono obbligatori e controllati mediante JS prima dell'invio del modulo. Si possono inserire al max. 10 artisti (minimo 1)
- a. La select delle categorie si autopopola dalla tabella categorie
 - b. La select dei luoghi mostra tutti i luoghi memorizzati nella tabella LUOGO (nome-città-prov) per una selezione facile e veloce
 - c. La select Arista mostra tutti gli artisti presenti nella corrispondente tabella del database

Pulsanti per l'invio dei dati o la cancellazione del form

7-CODIFICA IN UN LINGUAGGIO A SCELTA DI UN SEGMENTO SIGNIFICATIVO DELL'APP. WEB CHE CONSENTE L'INTERAZIONE CON LA BASE DEI DATI

LOGIN: Il form presenta due caselle di testo ed un pulsante. I dati vengono spediti solo se i campi non sono vuoti . Il file di controllo in PHP preleva i dati dal form ed interroga il database. Se l'utente viene trovato si avvia una sessione e si memorizzano i dati dell'utente nella sessione, altrimenti si è reindirizzati ad una pagina di errore

File: form_login.html (si omette per brevità la parte dei controlli e tutto il resto dell'HTML).

```
<form action="controllo_login.php" name="form_login" id="id_login" method="post"
onSubmit="return controlloLogin();" >
Utente: <input type="text" name="user" value="" />
Password: <input type="password" name="pw" value="" />
<input type="submit" name="accedi" value="Login" /><p> <br> </p>
</form>
```

File: controllo_login.php (si omette per brevità una parte dei controlli)

```
<?php
$username=$_POST['user']; //Recupera dati dal form
$password=$_POST['pw'];
//Interroga il database per verificare l'utente
$host="localhost";
$user="root";
$password="";
$conn=mysql_connect($host,$user,$pw) or die("connessione impossibile");
$dbname="eventi";//Definizione del database
mysql_select_db($dbname,$conn);
$sql="SELECT * FROM utenti where nickname='$username' AND password='$password'";
$risposta=mysql_query($sql)or die(mysql_error ());
$num_righe= mysql_num_rows($risposta);
if($num_righe!=1){
header('Location: view.php?id=6');//pagina gestione "Errore!";
exit;
```

```
}  
//Se Utente autenticato crea la sessione  
if(!isset($_SESSION['start'])) {session_start(); //creo la sessione}  
$_SESSION['start']=true;  
$_SESSION['datiUtente']=mysql_fetch_array($risposta);  
header('Location: view.php');//Pagina utente autenticato  
exit;  
?>
```

SOLUZIONE SECONDA PARTE

I PUNTO-INTEGRAZIONE SCHEMA CONCETTUALE PER GESTIONE BANNER PUBBLICITARIO COLLEGATO A PIU' CATEGORIE ED ALLE PAGINE VISITATE DAGLI UTENTI (INTERESSI)

Si può aggiungere una nuova entità (BANNER) collegata con l'entità CATEGORIE da una relazione N-N

Lo schema logico sarà il seguente:

BANNER (id, testo, link, immagine) – Contiene i riferimenti della pubblicità

CATEGORIA_BANNERE (id_categoria, id_banner) – Realizza la relazione N-N tra le due entità

La personalizzazione delle pubblicità può avvenire mediante l'uso dei cookie. Il sistema invia all'utente uno o più cookie nel quale memorizza informazioni circa le pagine visitate (e quindi gli interessi e le categorie di riferimento). Ad ogni nuova pagina richiesta analizza il cookie e seleziona (in base alla preferenze rilevate dal cookie) uno dei banner disponibili in quel contesto in modalità casuale (random).

II PUNTO-LAYOUT GRAFICO COMUNE A TUTTE LE PAGINE WEB CON PARZIALE CODIFICA DI ALCUNE PARTI

Come detto in premessa, tutte le pagine del sito avranno una struttura simile alla seguente:

Intestazione: Logo, nome sito, ricerca globale, slide show		
Barra di navigazione: HOME – LOGIN- REGISTRATI-EVENTI-CONTATTI-COLLABORA CON NOI		
Colonna sinistra MENU UTENTE LOGIN LINK INFO VARIE	Colonna centrale (corpo principale) NEWS – ARTICOLI- RISULTATI DELLE RICERCHE- VARI FORM INSERIMENTI-FOTO-ECC	Colonna destra
Pie' pagina: MAPPA SITO, CREDITS, LINK VARI, WEBMASTER, POSTA		

Ogni pagina del sito avrà una struttura HTM/CSS come quella riportata sotto.

Pagina HTML

```
<html>
<head>
<title>Eventi Live</title>
<link rel="stylesheet" type="text/css" href="css/style.css" />
<script type="text/javascript" src="js/controlli.js"></script>
</head>
```



```
<body>
<div id="main_container">
  <div id="header">
    INTESTAZIONE: LOGO, NOME DEL SITO, EVENTUALE SLIDE SHOW CON EVENTI FOTO DI
    EVENTI IMMINENTI, RICERCA GLOBALE NEL SITO
  </div>
  <div id="main_content">
    CORPO CENTRALE. CONTIENE: BARRA DI NAVIGAZIONE ORIZZONTALE, LAYOUT A TRE COLONNE
    (COLONNA SX, CORPO CENTRALE, COLONNA DX)
    <div id="menu_tab">
      BARRA DI NAVIGAZIONE CON I LINK
    </div>
    <!-- FINE BARRA MENU ORIZZONTALE -->
    <div class="left_content">
      CONTENUTI COLONNA SINISTRA
    </div>
    <!-- FINE COLONNA SINISTRA -->
    <div class="center_content">
      CONTENUTI COLONNA CENTRALE (CORPO PRINCIPALE DEI CONTENUTI)
    </div>
    <!-- FINE DEL CORPO CENTRALE -->
    <div class="right_content">
      CONTENUTI COLONNA DESTRA
    </div>
    <!-- fine colonna sinistra -->
  </div>
  <!-- fine del main content -->
  <div class="footer">
    CONTENUTI PIE' PAGINA (CREDENZIALI, WEB MASTER, MAPPA SITO, RIFERIMENTI, ECC)
  </div>
<!-- fine del main_container -->
</body>
</html>
```

Mentre il file style.css che realizza il layout potrebbe essere simile a questo.

```
body
{
background: url(images/bg.jpg) no-repeat #fff center top;
padding:0;
font-family:Arial, Helvetica, sans-serif;
font-size:11px;
margin:0px auto auto auto;
color:#000;
}
#main_container{
width:1000px;
height:auto;
margin:auto;
padding:0px;
}
#header{
width:1000px;
height:173px;
background: url(images/header_bg.jpg) no-repeat center;
background-position:0px 0px;
margin:auto;
}
#main_content{
```

```
clear:both;
}
/*----- menu tab-----*/
#menu_tab{
width:1000px;
height:50px;
background-color:#fff;
}
/*-----left_content-----*/
.left_content{
width:196px;
float:left;
}
/*center content-----*/
.center_content{
width:585px;
float:left;
padding:5px 10px;
}
/*-----right content-----*/
.right_content{
width:196px;
float:left;
padding:5px 0 0 0;
}
/*-----footer-----*/
.footer{
width:1000px;
clear:both;
height:65px;
background: url(images/footer_bg.gif) repeat-x top;
}
```

III PUNTO-TRASFORMAZIONE TABELLA IN 3FN

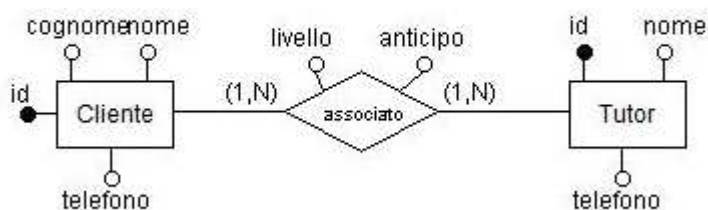
Supponendo come chiave primaria: nome,cognome e telefono, gli attributi: Livello, Tel-tutor e Tutor non dipendono esclusivamente dalla chiave primaria ma, ad esempio, l’attr. Tel-tutor dipende dal Tutor (due non-chavi) così come il Livello è legato al tipo di corso e non alla persona.

Per essere in 3NF la tabella andrebbe scomposta così:

CLIENTE(ID, COGNOME, NOME, TEL) → Contiene i clienti dei vari corsi

ASSOCIATO (LIVELLO, ANTICIPO, ID_CLIENTE, ID_TUTOR) → Associazione cliente-tutor. Rel. N-N

TUTOR (ID, NOME,TEL) → Contiene l’elenco dei tutor



IV PUNTO-CARDINALITA' SIGNIFICATO E CASISTICA

Per ogni entità partecipante ad una relazione viene specificata una **cardinalità di relazione**. Essa è una coppia di numeri naturali che specifica il numero minimo e massimo di istanze di relazione a cui una istanza dell'entità può partecipare. E' possibile indicare con la costante N un numero generico maggiore di uno quando la cardinalità non è nota con precisione. Nel diagramma ER, la cardinalità di relazione di una entità etichetta l'arco che lega l'entità alla relazione.

Definizione.: La **cardinalità** dell'associazione indica il numero di istanze dell'entità di arrivo che si associano all'istanza dell'entità di partenza.

La cardinalità può essere **a uno** oppure **a molti** e pertanto le associazioni tra due entità si classificano nei seguenti tipi:

- Associazione **uno a uno**, indicata con **1:1**
- Associazione **uno a molti**, indicata con **1:N**
- Associazione **molti a molti**, indicata con **N:N**

Un'associazione si dice **uno a uno** e si indica con **1:1** quando ogni istanza della prima entità si deve associare ad una sola istanza della seconda entità e viceversa

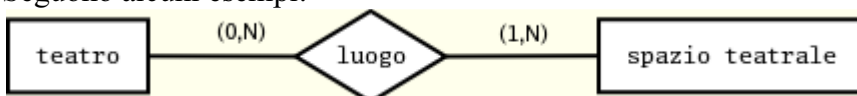
Un'associazione si dice **uno a molti** e si indica con **1:N** quando ogni istanza della prima entità si può associare a una o più istanze della seconda entità, mentre ad ogni istanza della seconda entità si deve associare una sola istanza della prima.

Un'associazione si dice **molti a molti** e si indica con **N:N** se ad ogni istanza della prima entità si possono associare una o più istanze della seconda entità e ad ogni istanza della seconda entità si possono associare una o più istanza della prima.

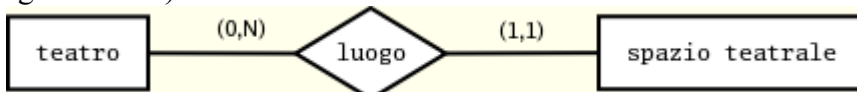
La partecipazione di un'entità all'associazione può essere **obbligatoria** quando il legame tra le entità **deve** essere sempre presente, **facoltativa(opzionale)** quando **può** essere presente

Parte facoltativa del punto IV

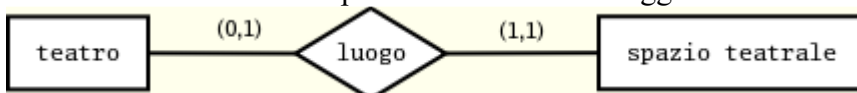
Seguono alcuni esempi:



In questo caso un teatro ha nessun, uno o molti spazi teatrali e uno spazio teatrale ha uno o più teatri associati. Questo tipo di relazione in cui entrambe le cardinalità massime sono maggiori di uno si chiama **molti a molti**. La partecipazione dell'entità teatro si dice **opzionale** (cardinalità minima uguale a zero) e la partecipazione dell'entità spazio teatrale si dice **obbligatoria** (cardinalità minima uguale a uno).



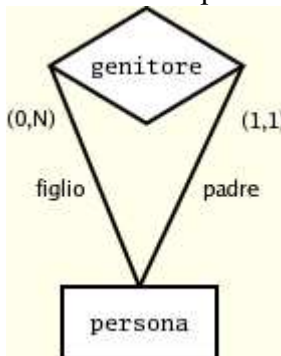
In questo caso un teatro ha nessun, uno o molti spazi teatrali e uno spazio teatrale ha esattamente un teatro associato. Dunque non vi sono spazi condivisi da più teatri. Questo tipo di relazione in cui una cardinalità massima è pari a uno e l'altra è maggiore di uno si chiama **uno a molti**.



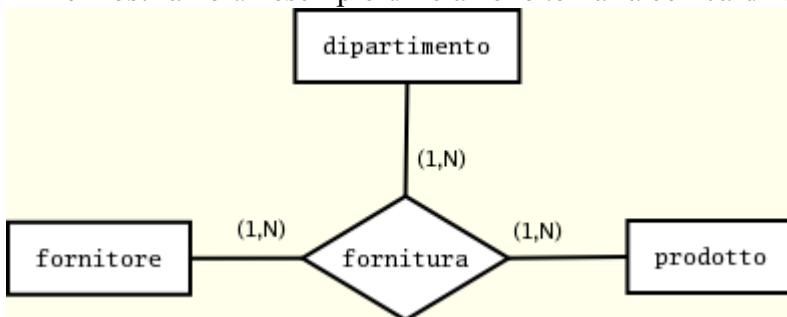
In questo caso un teatro ha nessun o uno spazio teatrale e uno spazio teatrale ha esattamente un teatro associato. Dunque non vi sono teatri con più spazi e neppure spazi condivisi da più teatri.

Questo tipo di relazione in cui entrambe le cardinalità massime sono uguali a uno si chiama **uno a uno**.

Di seguito proponiamo un esempio di relazione ricorsiva con ruoli e cardinalità. Ogni persona ha esattamente un padre e zero o più figli:



Infine mostriamo un esempio di relazione ternaria con cardinalità:



Si ricordi che la cardinalità di una entità si riferisce al numero di istanze della relazione a cui quella entità può partecipare. Ad esempio, nel caso di fornitore, la cardinalità si riferisce al numero di *coppie* prodotto e dipartimento che il fornitore rifornisce. Dunque una cardinalità massima pari a uno per l'entità fornitore si ha quando un fornitore rifornisce un solo dipartimento di un solo prodotto.

Regole di lettura:

Ogni <nome entità di partenza>**può | deve**<nome associazione>**un solo | uno o più**<nome entità di arrivo>

Es.: **Ogni** fornitore **deve** fornire **uno o più** prodotti

Regole di derivazione delle associazioni da modello E/R a modello logico relazionale:

- Ogni entità diventa una relazione
- L'associazione **uno a uno** viene tradotta in un'unica relazione che contiene gli attributi della prima e della seconda entità
- L'associazione **uno a molti** viene tradotta aggiungendo agli attributi dell'entità **a molti** la chiave dell'entità **a uno** e gli attributi dell'associazione
- L'associazione **molti a molti** viene tradotta mantenendo inalterate le due entità coinvolte dell'associazione ed aggiungendo una terza relazione (tabella) contenente le chiavi delle due entità e gli eventuali attributi dell'associazione

Qui, volendo, si potevano fare eventuali esempi esplicativi