

# ESAME DI MATURITA' 2015

## Seconda prova scritta indirizzo Informatica Abacus

### Informatica

#### Premessa.

La traccia comprende due parti.

La prima, in linea con la tradizione, chiede l'implementazione di un Data Base, partendo da una situazione reale.

La seconda è strutturata secondo quesiti che riguardano la gestione di database e l'implementazione di pagine web per accedervi.

Nel complesso la traccia è chiara e completa di tutte le indicazioni necessarie.

Qualche perplessità sulla seconda parte che potrebbe dare l'impressione di essere, almeno in parte, ridondante, fino a sembrare una terza prova inserita nella seconda.

Prima di passare allo sviluppo della soluzione precisiamo che:

- quella proposta è "una", non "la" soluzione
- il taglio dato è scolastico, in linea con i programmi curriculari.

#### **ANALISI**

Prima parte.

Nell'ambito di una web community si vuole creare uno strumento di condivisione di notizie e giudizi relativi ad eventi culturali. Utilizzando un database, ogni singolo utente registrato, può aggiungere notizie di eventi o commenti ed essere aggiornato su accadimenti relativi a specifiche aree di suo interesse previsti nella provincia di appartenenza tramite posta elettronica.

Gli utenti non registrati possono accedere comunque alle notizie, ma solo per visualizzarle.

Seconda parte.

Il primo quesito parla della possibilità di inserire pubblicità collegandola, in modo mirato, a ai contenuti visitati ed alle preferenze degli utenti. Questa specificazione implica che ad ogni accesso il server che ospita la community forzi un cookie nel computer di chi si collega.

Possiamo quindi pensare che ogni pagina del sito debba richiedere l'attivazione dei cookie e che sul computer client gli stessi debbano essere abilitati.

Per gestire una situazione di questo tipo risulta conveniente l'uso del linguaggio PHP.

Per il secondo quesito la traccia chiede di progettare sostanzialmente un layout comune a tutte le pagine .

Per il terzo quesito, la tabella andrebbe corredata di ulteriori informazioni, almeno di contesto.

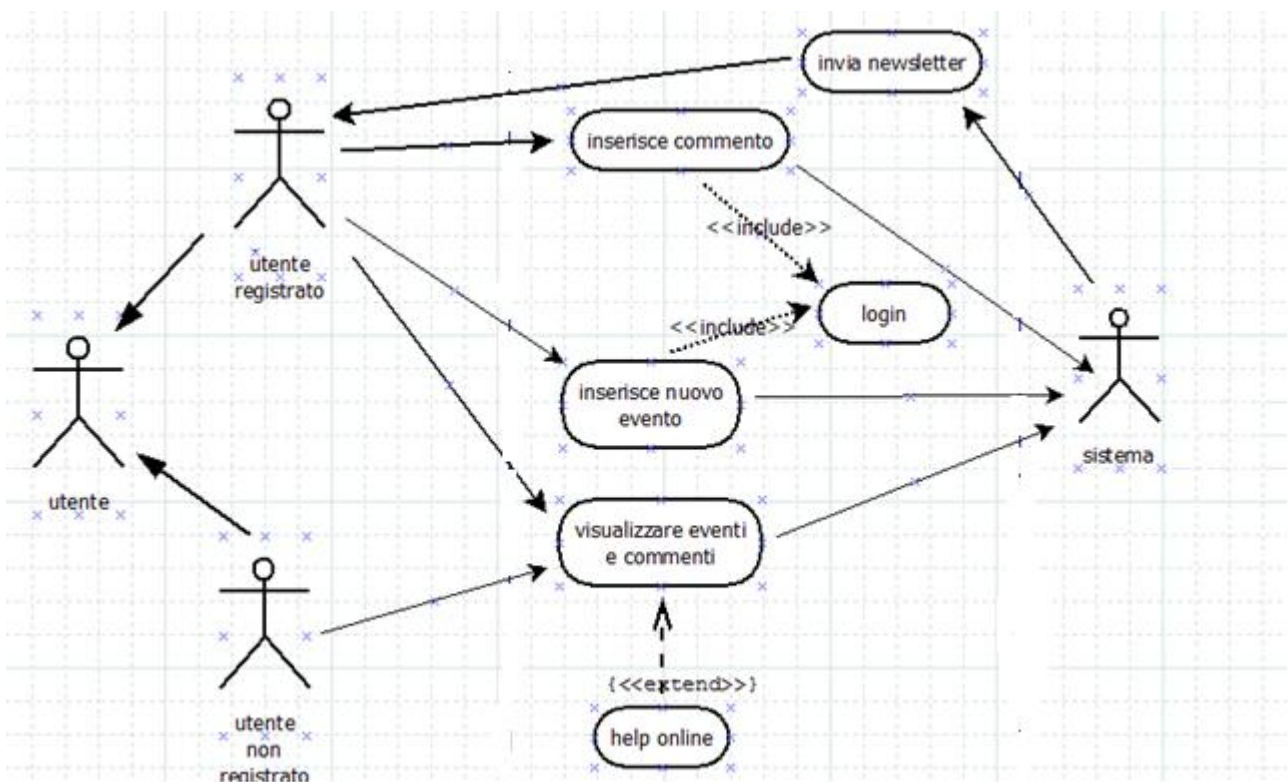
Sembra si tratti di una situazione relativa ad un corso,, articolato su due livelli, i cui iscritti sono seguiti da un tutor.

Il quarto quesito è solo teorico.

## Rappresentazione del sistema

### Schema funzionale.

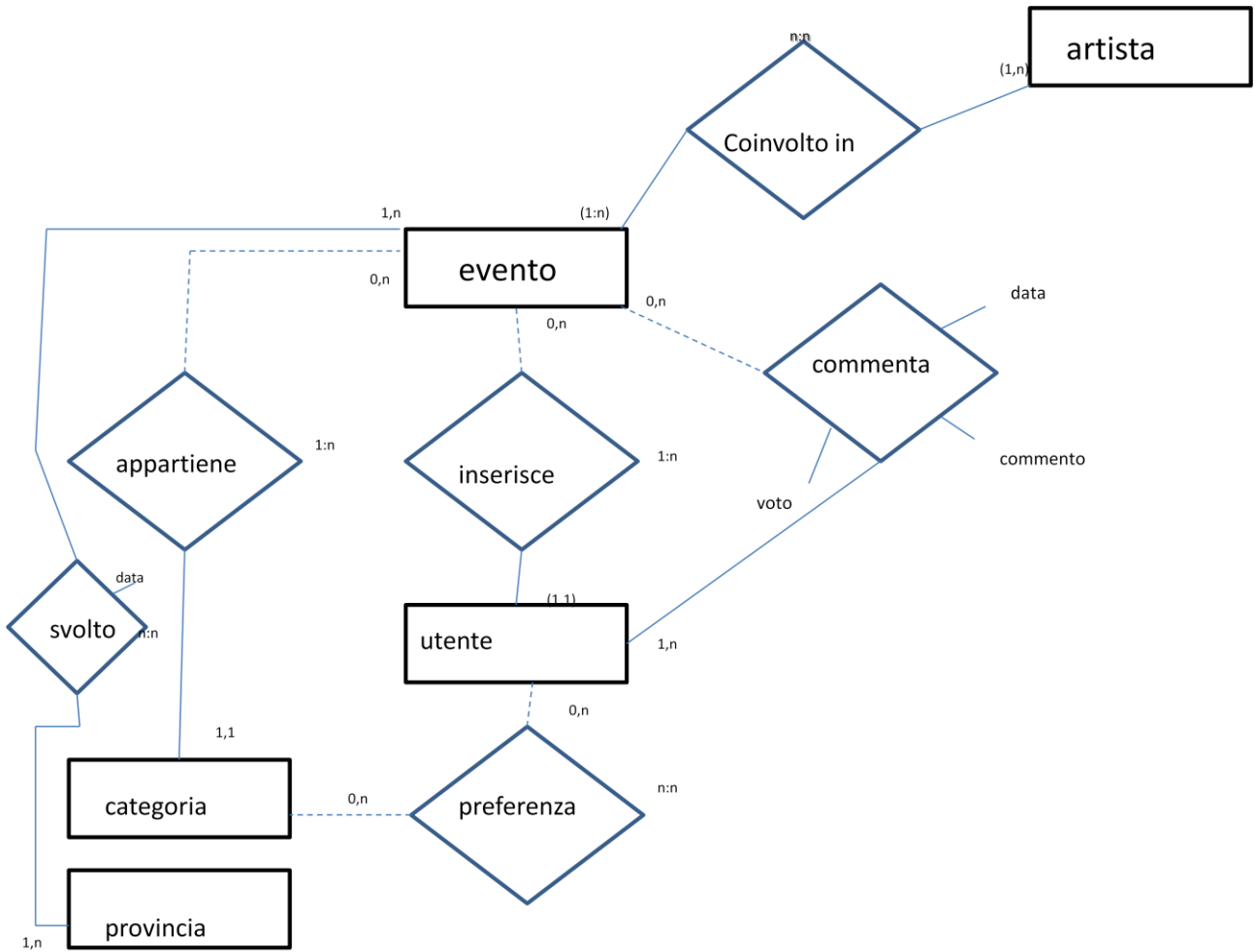
Utilizziamo un strumento di rappresentazione standard, legato al linguaggio UML, anche per suggerirne l'uso nella didattica. Nella figura che segue è riportato uno " use case" sintetico, riferito al sistema in studio.



Le stesse informazioni, vengono solitamente fornite sotto forma di moduli o blocchi funzionali, raggruppati per elenchi di manipolazione ed interrogazione.

## Modello E-R

Estrapolando dalla traccia i termini essenziali è possibile individuare le entità. Utilizzando poi, le informazioni del testo, si ricavano abbastanza agevolmente associazioni ed attributi. Per semplicità trascuriamo l'azione d'invio della newsletter agli utenti registrati



## Schema logico

Riportiamo di seguito le tabelle necessarie

UTENTE

<u>id</u>	nickname	nome	cognome	email	login	password
-----------	----------	------	---------	-------	-------	----------

CATEGORIA\_INTERESSANTE

categoria	utente
-----------	--------

CATEGORIA

<u>id</u>	nome
-----------	------

EVENTO

<u>id</u>	titolo	categoria	luogo	utente	data
-----------	--------	-----------	-------	--------	------

ARTISTA

<u>id</u>	nome_d'arte	nome	cognome
-----------	-------------	------	---------

COINVOLTO\_IN

artista	evento
---------	--------

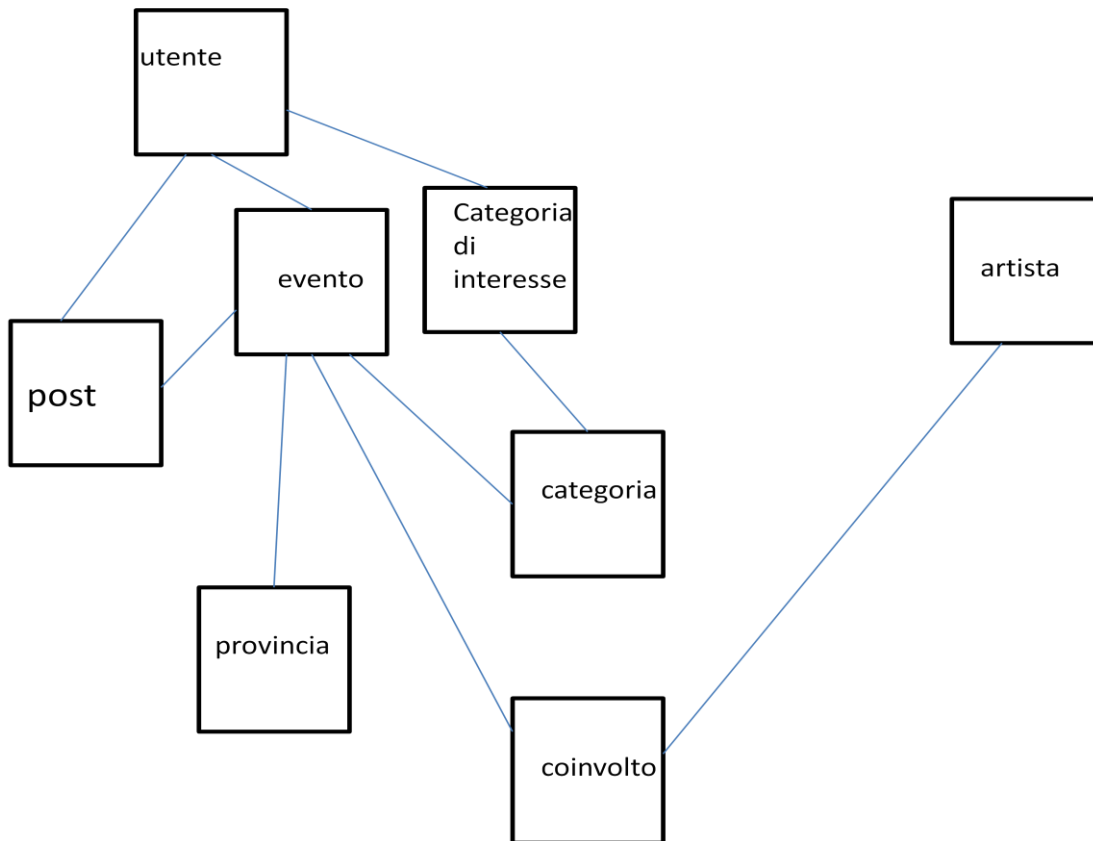
POST

<u>id</u>	utente	evento	commento	voto	data/ora
-----------	--------	--------	----------	------	----------

PROVINCIA

<u>luogo</u>	provincia
--------------	-----------

## Sintesi relazione tabelle



Per completezza, anche se risulta più complesso, riportiamo i collegamenti tra le tabelle con le chiavi coinvolte.

UTENTE(id, nickname, nome, cognome, email, login, password)

CATEGORIA\_INTERESSANTE(categoria, utente)

CATEGORIA(id, nome)

EVENTO(id, titolo, categoria, luogo, utente, data)

ARTISTA(id, nome\_d'arte, nome, cognome)

COINVOLTO\_IN(artista, evento)

POST(id, utente, evento, commento, voto, data/ora)

PROVINCIA(luogo, provincia)

#### **Punto 4**

```
CREATE TABLE POST
(id Char(7),
utente Char(7),
evento Char(7),
commento Char(150)
voto Integer,
data_ora Timestamp,
PRIMARY KEY (id),
FOREIGN KEY (utente) REFERENCES utente(id)
ON DELETE RESTRICT
ON UPDATE CASCADE,
FOREIGN KEY (evento)
REFERENCES evento(id)
ON DELETE CASCADE,
ON UPDATE CASCADE);
```

#### **Punto 5**

a.

```
SELECT titolo, data, luogo
FROM evento, provincia
WHERE evento.luogo = provincia.luogo
AND data < CURRENT DATE
ORDER BY provincia.provincia;
```

b.

```
SELECT nickname, nome, cognome, email
FROM utente
```

```
WHERE id NOT IN  
  
    (SELECT DISTINCT post.utente  
  
     FROM post);
```

c.

```
SELECT evento.id, evento.titolo, AVG(post.voto), categoria.nome  
  
FROM evento, post, categoria  
  
WHERE evento.id = post.evento AND categoria.id = evento.categoria  
  
GROUP BY evento.id  
  
ORDER BY categoria.nome, evento.titolo;
```

d.

```
CREATE VIEW eventi_per_utente AS  
  
SELECT utente, COUNT(utente) AS conteggio  
  
FROM evento;
```

```
SELECT utente.nickname, utente.nome, utente.cognome, utente.email  
  
FROM utente, eventi_per_utente  
  
WHERE utente.id = eventi_per_utente.utente AND eventi_per_utente.conteggio =  
  
    (SELECT MAX(conteggio)  
  
     FROM eventi_per_utente);
```

## Punto 6

Una semplice ( per questioni di tempo e spazio) home page del sito è riportata di seguito

Intestazione: <b>community web</b>		
eventi >>		
<b>Ricerca</b> <b>Aggiungi</b> <b>Commenti</b> <b>Registrazione</b>	Riferimenti e contenuti alla voce scelta	<b>Area riservata agli iscritti</b> Nome <input type="text"/> Password <input type="text"/> <input type="button" value="Login"/>
Ricerca Aggiungi Commenti Registrazione		

## Punto 7.

Considerato che la traccia prevede l'accesso di utenti registrati proponiamo un form di login con due campi di input e un pulsante di invio (login):

```
<html>
  <body>
    <form action="autenticazione.php" method="post">
      Nome: <input type="text" name="user"><br>
      Password: <input type="text" name="pw"><br>
      <input type="submit" name="submit" value="Login">
    </form>
  </body>
</html>
```

Quando l'iscritto compila il modulo sopra e fa clic sul pulsante di invio, i dati del form vengono passati per l'elaborazione al file PHP denominato "autenticazione.php". I dati del modulo vengono inviati con il metodo HTTP POST.

La casella di testo di nome passa il suo valore alla pagina autenticazione.php che recupera il contenuto della variabile utilizzando \$user e \$pw:

```
<?php
  // define variables and set to empty values
```



```

$user = $pw = " ";
if ($_SERVER["REQUEST_METHOD"] == "POST")
{
    $name = test_input($_POST["user"]);
    $pwd = test_input($_POST["pw"]);
}
}
?>

```

I valori delle variabili user e pw vengono utilizzati di seguito nello script di connessione che si collega al server contenente il database degli eventi.

L'oggetto per la connessione ha una struttura del tipo:

```
$ mysqli = new mysqli("username", "user", "pw", "database");
```

nel caso particolare possiamo ipotizzare una specificazione del tipo:

```
$ hostname= "174.1.4.7";
$ database = "eventi";
```

```
$ mysqli = new mysqli( $hostname,$user, $pw,"database");
```

ne deriva il codice php che segue

```

?php
// connessione a MySQL con l'estensione MySQLi

$ hostname= "174.1.4.7";
$ database = "eventi";

$ mysqli = new mysqli( $hostname,$user, $pw,"database");

// verifica dell'avvenuta connessione
if (mysqli_connect_errno()) {

    echo "Errore : ".mysqli_connect_error();
    // interruzione delle esecuzioni i caso di errore
    exit();

}
else {
    // notifica in caso di connessione attiva
    echo "Connessione avvenuta con successo";
}

```

**SECONDA PARTE**

**Quesito 1**



```
list-style-type: none;

padding: 0;

margin: 0;

position: absolute;

top: 2em;

left: 1em;

width: 9em }

h1 {

font-family: Helvetica, Geneva, Arial,

SunSans-Regular, sans-serif }

ul.navbar li {

background: white;

margin: 0.5em 0;

padding: 0.3em;

border-right: 1em solid black }

ul.navbar a {

text-decoration: none }

a:link {

color: blue }

a:visited {

color: purple }

address {

margin-top: 1em;

padding-top: 1em;

border-top: thin dotted }
```

Salviamo il file con nome esempiostile.css nella cartella di lavoro.

Successivamente, in ogni pagina HTML, dovremo aggiungere, come intestazione, un codice del tipo:

```
<html>  
  
<head>  
  
  <title>Il mio stile</title>  
  
  <link rel="stylesheet" href="esempiostile.css">  
  
</head>
```

```
<body>
```

```
.....Segue corpo pagina
```

### **Quesito III**

La tabella è già in prima forma normale (1NF) in quanto rispetta i requisiti fondamentali del modello relazionale, che sono:

tutte le righe della tabella contengono lo stesso numero di colonne;

gli attributi rappresentano informazioni elementari;

i valori che compaiono in una colonna sono dello stesso tipo, cioè appartengono allo stesso dominio;

ogni riga è diversa da tutte le altre, cioè non ci possono essere due righe con gli stessi valori nelle colonne;

l'ordine con il quale le righe compaiono in tabelle è irrilevante.

Una tabella è in seconda forma normale (2NF) quando, oltre ad essere ovviamente in 1NF, possiede campi non chiave che dipendono funzionalmente dall'intera chiave primaria e non da una parte di essa.

Una tabella è in terza forma normale (3NF) quando, oltre ad essere ovviamente in 2NF, possiede campi non chiave che dipendono direttamente dalla chiave, cioè non possiede attributi non-chiave che dipendono da altri attributi non-chiave.

Di conseguenza uno schema equivalente e normalizzato potrebbe esser il seguente:

Tabella CORSISTA

<u>id</u>	nome	cognome	telefono
-----------	------	---------	----------

Tabella CORSO

<u>id</u>	nome	idlivello
-----------	------	-----------

Tabella LIVELLO\_CORSO

<u>id</u>	livello
-----------	---------

Tabella TUTOR

<u>id</u>	nome	cognome	telefono	idcorso
-----------	------	---------	----------	---------

Tabella CORSISTA\_FREQUENTA\_CORSO

idcorsista	idcorso	anticipo_versato
------------	---------	------------------

#### Quesito IV

Nella formalizzazione di uno schema concettuale, le associazioni tra entità sono caratterizzate dal grado, che rappresenta il numero di entità che partecipano all'associazione e possono essere quindi: associazioni unarie (di grado uno), associazioni binarie (di grado due), associazioni n-arie (di grado maggiore a 2).

Per ogni entità che partecipa ad un'associazione possono essere definiti i vincoli di cardinalità che rappresentano il numero massimo e minimo di istanze dell'associazione a cui un'entità può partecipare.

In generale, la cardinalità può essere un qualunque numero intero non negativo, con l'ovvio vincolo che la cardinalità massima sia maggiore delle minima. Tuttavia, i valori più comunemente utilizzati sono 1 e 0 per la minima e 1 e n per la massima (n indica un qualunque numero intero maggiore di 1).

Il significato di questi valori è il seguente:

se la cardinalità minima di una classe rispetto ad un'associazione è 0, allora la partecipazione della classe all'associazione è opzionale, cioè possono esistere delle istanze della classe che non partecipano ad alcuna istanza dell'associazione;

- se la cardinalità minima di una classe rispetto ad un'associazione è 1, allora la partecipazione della classe all'associazione è obbligatoria, cioè non può esistere un'istanza della classe che non partecipa ad almeno un'istanza dell'associazione
- se la cardinalità massima di una classe rispetto ad un'associazione è 1, allora ogni istanza della classe potrà partecipare al massimo ad un'istanza dell'associazione. Se, in aggiunta, la cardinalità minima è 1, allora ogni istanza della classe parteciperà ad una e una sola istanza dell'associazione.
- se la cardinalità massima di una classe rispetto ad un'associazione è n, allora non esiste limite al numero massimo di istanze dell'associazione a cui ogni istanza della classe può partecipare. Se la cardinalità minima è 0, allora ogni istanza della classe parteciperà ad un numero qualsiasi di istanze dell'associazione.

Alessandra Valpiani Informatica IIS LAGRANGE Milano

Domenico Capezzuto Sistemi IIS LAGRANGE Milano