

## **Ministero dell'Istruzione, dell'Università e della Ricerca**

### **I044 – ESAME DI STATO DI ISTRUZIONE SECONDARIA SUPERIORE Sessione straordinaria 2017**

Indirizzo: ITIA - INFORMATICA E TELECOMUNICAZIONI

ARTICOLAZIONE INFORMATICA

Tema di: INFORMATICA – Tipologia C

Il candidato (che potrà eventualmente avvalersi delle conoscenze e competenze maturate attraverso esperienze di alternanza scuola-lavoro, stage o formazione in azienda) svolga la prima parte della prova e due tra i quesiti proposti nella seconda parte.

#### **PRIMA PARTE**

Un'associazione culturale intende organizzare una manifestazione pubblica che prevede la presentazione di uno o più "contributi" da parte di "espositori".

Coloro che intendono diventare espositori devono registrarsi come utenti presso la piattaforma online riservata alla manifestazione. L'interessato fornisce i propri dati di registrazione: username di accesso al sistema, una password, cognome, nome, indirizzo di posta elettronica, numero di telefono, qualifica (scelta tra: professionista del settore, amatore, esperto non professionista) e allega un file in formato pdf con un curriculum vitae in formato europeo.

Gli utenti registrati possono poi compilare una o più schede di candidatura per proporre un proprio contributo espositivo: per ciascuna scheda di candidatura deve essere fornito un titolo (testo fino a 255 caratteri), una sintesi descrittiva (testo fino a 2000 caratteri), un'immagine allegata, un URL che rilancia a pagine di approfondimento, una o più categorie di classificazione scelte fra quelle proposte dagli organizzatori.

Il personale dell'associazione esamina le schede pervenute, registrando sulla piattaforma la loro accettazione o il rifiuto. A conclusione di questa fase, un'apposita funzione del sistema di gestione provvede ad inviare agli interessati una email di accettazione o di rifiuto per ciascuna candidatura presentata. Le candidature accettate diventano così "contributi" che saranno esposti durante la manifestazione.

Sulla piattaforma online, una sezione ad accesso libero presenta i contributi accettati, nella forma di un elenco con titolo e cognome/nome dell'espositore. Selezionando ciascuna riga di tale elenco è possibile chiedere una vista di dettaglio di ogni singolo contributo, dove viene mostrato ogni elemento noto del contributo stesso e il curriculum dell'espositore.

Per esigenze logistiche, la manifestazione viene organizzata raggruppando i diversi espositori in aree, ciascuna delle quali ha una capienza massima.

Il candidato, fatte le opportune ipotesi aggiuntive, sviluppi:

1. un'analisi della realtà di riferimento, giungendo alla definizione di uno schema concettuale della base di dati che, a suo motivato giudizio, sia idoneo a gestire la realtà presentata;
2. il relativo schema logico;
3. le seguenti interrogazioni espresse in linguaggio SQL:
  - a. elenco degli espositori, in ordine alfabetico per cognome e nome con l'area a cui sono assegnati;
  - b. elenco delle esposizioni di una certa categoria di classificazione, con indicazione del titolo, cognome e nome dell'espositore e area di esposizione;
  - c. l'elenco delle categorie di classificazione, in ordine decrescente per il numero di espositori;
4. il progetto di massima della struttura funzionale dell'applicazione Web, realizzando, con appropriati linguaggi a scelta sia lato client che lato server, la parte che consente la visualizzazione dei dati di un contributo nella sezione ad accesso libero.

## SECONDA PARTE

- I. In relazione al tema sviluppato nella prima parte, e considerando che ogni area ha una propria capienza massima che non può essere superata per motivi di sicurezza, l'accesso dei visitatori alle diverse aree viene organizzato in turni. Il candidato integri la base di dati (schema concettuale e logico) in modo da gestire on-line la prenotazione dei visitatori. Ogni visitatore registra un proprio profilo e poi può prenotare una o più aree e turni per la visita. Il sistema di registrazione impedirà la scelta di un'area e turno per i quali è già stata raggiunta la capienza limite. Il candidato realizzi, con appropriati linguaggi a scelta sia lato client che lato server, la parte dell'applicazione Web che consente ad un visitatore registrato di svolgere le operazioni di prenotazione.
- II. In relazione al tema sviluppato nella prima parte, si consideri che ogni utente registrato ha la possibilità di creare, rivedere, modificare, cancellare le proprie schede di candidatura fino ad una data limite prefissata, dopo di che ne potrà solo prendere visione. Il candidato produca il layout grafico della porzione dell'applicazione Web che consente agli utenti registrati di svolgere le operazioni specificate. Descriva poi la struttura di una guida on-line a supporto degli utenti registrati per illustrare tali funzionalità, scrivendo anche un esempio significativo dei contenuti.
- III. Il candidato presenti una panoramica delle tecnologie di sua conoscenza per lo sviluppo di applicazioni Web dinamiche. Approfondisca le differenze tra tali tecnologie, nelle funzionalità ed utilizzi tipici, e le esemplifichi, facendo riferimento al seguente caso concreto: in una applicazione Web viene richiesto di compilare tre campi di residenza con Regione, Provincia e Comune, che vengono impostati tramite tre listbox in cui la seconda e la terza presentano valori diversificati in funzione di quanto selezionato rispettivamente nella prima e nella seconda (es: se come regione viene selezionata Liguria dall'elenco delle Regioni, la listbox Provincia dovrà mostrare i valori Imperia, Savona, Genova, La Spezia).
- IV. Il candidato illustri le attività tipiche di un amministratore di una base di dati, corredandole con esempi significativi di comandi SQL specifici per lo svolgimento delle funzioni a lui riservate.

---

Durata massima della prova: 6 ore.

È consentito l'uso di manuali tecnici e di calcolatrici tascabili non programmabili.

È consentito l'uso del dizionario bilingue (italiano-lingua del paese di provenienza) per i candidati di madrelingua non italiana.

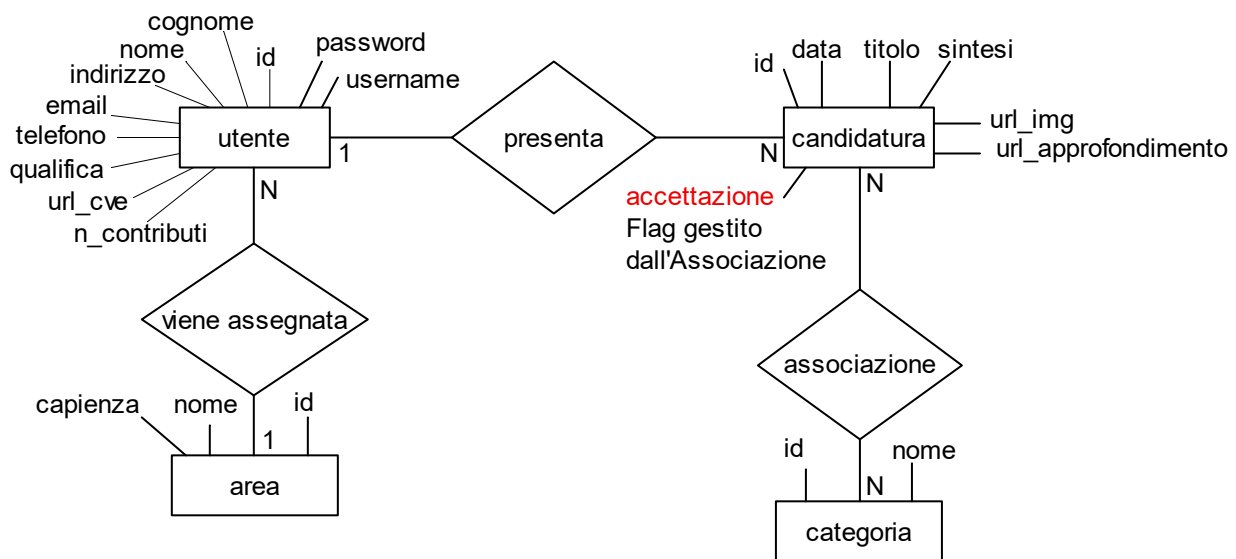
Non è consentito lasciare l'Istituto prima che siano trascorse 3 ore dalla dettatura del tema.

## Soluzione prima parte

### Schema concettuale (Diagramma E/R)

#### Ipotesi di lavoro

- a) la qualifica dell'utente che si candida ad esporre uno o più contributi viene considerata come attributo e non come entità. L'attributo "qualifica" può assumere i seguenti valori: P (Professionista del settore), A (Amatore), E (Esperto non professionista)



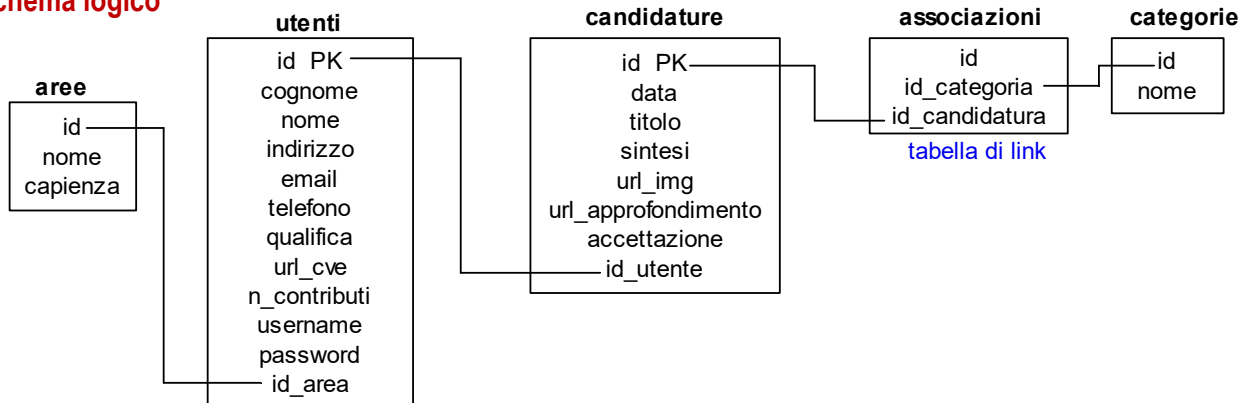
#### Letture del diagramma E/R

- Ogni utente registrato presenta N schede di candidatura, ogni scheda è presentata da un utente
- Ogni scheda di candidatura viene associata dall'utente a N categorie di classificazione, ad ogni categoria sono associate N schede di candidatura
- Ad ogni utente con una o più candidature accettate (espositore), viene assegnata un'area, in ogni area vengono assegnati N espositori

#### Chiarimenti

- Riguardo alle candidature, "Il personale dell'associazione esamina le schede pervenute, registrando sulla piattaforma la loro accettazione o il rifiuto"; in pratica gestisce in modo esclusivo il flag "accettazione" assegnandogli il valore "A" se la candidatura viene accettata o il valore "R" se la candidatura viene rifiutata. Quando la candidatura viene accettata:
  - a) la candidatura diventa un "contributo" o, come viene chiamata anche nella traccia, un'esposizione
  - b) l'utente diventa un "espositore"
  - c) "n\_contributi", ovvero il numero di contributi dell'espositore, viene automaticamente incrementato di 1
- In base alla capienza massima di ciascuna area e, verosimilmente, tenendo conto del numero e della natura dei contributi di ciascun espositore, il personale dell'Associazione assegna a ciascun espositore un'area specifica della manifestazione

## Schema logico



## Definizione delle relazioni in linguaggio SQL

### Struttura delle tabelle

Si fa riferimento al database MySql

Nome del database: **contributi**

tabella	campi	key	tipo	lung.	descrizione/note
aree	id	PK	Int		Contatore auto_increment
	nome		VarChar	30	Esempio: Salone A
	capienza		Int		Numero di visitatori
categorie	id	PK	Int		Contatore auto_increment
	nome		VarChar	30	Esempio: Installazione artistica
utenti	id	PK	Int		Contatore auto_increment
	cognome		VarChar	30	
	nome		VarChar	30	
	indirizzo		VarChar	80	
	email		VarChar	30	
	telefono		VarChar	30	
	qualifica		Char	1	Può assumere i valori P,A,E
	url_cve		Text		Link al "Curriculum vitae europeo" in pdf
	n_contributi		Int		Numero contributi, ovvero numero candidature accettate.
	password		VarChar	32	Si utilizza la codifica MD5()
candidature	id	PK	Int		Contatore auto_increment
	data		Date		
	titolo		VarChar	255	
	sintesi		Text		
	url_img		Text		Link ad un'immagine allegata
	url_approfondimento		Text		Link ad un documento allegato o ad una pagina web
	accettazione		Char		Flag gestito dall'associazione. Vale "A" se la candidatura dell'utente viene accettata, "R" se viene rifiutata. Se accettazione="A" la candidatura diventa un contributo
	id_utente	FK	Int		Fa riferimento a utenti.id
associazioni Tabella di link	id	PK	Int		Contatore auto_increment
	id_categoria	FK	Int		Fa riferimento a categorie.id
	id_candidatura	FK	Int		Fa riferimento a candidature.id

## Query MySql per la creazione del database e delle tabelle

```
CREATE DATABASE contributi;
```

```
CREATE TABLE IF NOT EXISTS aree (  
id INT AUTO_INCREMENT PRIMARY KEY,  
nome VARCHAR(30),  
capienza Int);
```

```
CREATE TABLE IF NOT EXISTS categorie (  
id INT AUTO_INCREMENT PRIMARY KEY,  
nome VARCHAR(30));
```

```
CREATE TABLE IF NOT EXISTS utenti(  
id INT AUTO_INCREMENT PRIMARY KEY,  
cognome VARCHAR(40),  
nome VARCHAR(40),  
indirizzo VARCHAR(80),  
email VARCHAR(30),  
telefono VARCHAR(30),  
qualifica CHAR(1),  
url_cve TEXT,  
n_contributi INT,  
username VARCHAR(20),  
password VARCHAR(32),  
id_area INT,  
FOREIGN KEY (id_area) REFERENCES aree(id));
```

```
CREATE TABLE IF NOT EXISTS candidature (  
id INT AUTO_INCREMENT PRIMARY KEY,  
data DATE,  
titolo VARCHAR(255),  
sintesi TEXT,  
url_img TEXT,  
url_approfondimento TEXT,  
accettazione CHAR(1),  
id_utente INT,  
FOREIGN KEY (id_utente) REFERENCES utenti(id));
```

```
CREATE TABLE IF NOT EXISTS associazioni (  
id INT AUTO_INCREMENT PRIMARY KEY,  
id_categoria INT,  
id_candidatura INT,  
FOREIGN KEY (id_categoria) REFERENCES categorie(id),  
FOREIGN KEY (id_candidatura) REFERENCES candidature(id));
```

## Interrogazioni espresse in linguaggio SQL

Le query sono state testate utilizzando MySql e PhpMyAdmin di Wamp Server 3.3.0 64 bit

Elenco degli espositori, in ordine alfabetico per cognome e nome con l'area a cui sono assegnati

```
Select utenti.cognome As "Cognome",utenti.nome as "Nome",aree.nome as "Area"  
From utenti,aree  
Where aree.id=utenti.id_area  
AND utenti.n_contributi>0  
Order by utenti.cognome,utenti.nome
```

Nota: n\_contributi>0 indica che almeno una candidatura è stata accettata e che l'utente è un espositore

Cognome	Nome	Area assegnata
Labarca	Remo	Salone delle feste
Lavespa	Guido	Salone delle feste
Rossi	Mario	Corridoio
Verdi	Paolo	Scuderia

Elenco delle esposizioni di una certa categoria di classificazione, con indicazione del titolo, cognome e nome dell'espositore e area di esposizione

Scegliamo la categoria "Scultura"

```
Select titolo as "Titolo della scultura", utenti.cognome as "Cognome",  
utenti.nome as "Nome",aree.nome as "Area"  
From candidature,utenti,aree,categorie,associazioni  
Where candidature.id_utente=utenti.id  
And utenti.id_area=aree.id  
And associazioni.id_categoria=categorie.id  
And associazioni.id_candidatura=candidature.id  
And categorie.nome="Scultura"  
And accettazione="A"  
order by aree.nome
```

Nota: accettazione="A" indica che la candidatura è stata accettata ed è un contributo (un'esposizione)

Titolo della scultura	Cognome	Nome	Area
Testa di cavallo	Rossi	Mario	Corridoio
Bicicletta	Rossi	Mario	Corridoio
scultura per la pace	Labarca	Remo	Salone delle feste
Falco reale	Verdi	Paolo	Scuderia

Elenco delle categorie di classificazione, in ordine decrescente per il numero di espositori

Per semplificare una query che sarebbe complessa, creo una tabella temporanea con l'elenco delle candidature accettate (contributi), ciascuna con l'id utente (espositore) e il nome della categoria associata

Create table temp AS

```
Select candidature.id,candidature.id_utente,categorie.nome
```

```
From candidature,associazioni,categorie
```

```
where candidature.accettazione="A"
```

```
and candidature.id=associazioni.id_candidatura
```

```
and associazioni.id_categoria =categorie.id
```

id	id_utente	nome
1	1	Pittura
2	1	Pittura
3	2	Installazione artistica
4	3	Scultura
5	3	Scultura
6	4	Mosaico
7	4	Scultura
8	1	Scultura

Raggruppo le categorie in ordine decrescente per il numero di espositori

```
Select count(DISTINCT id_utente) as "Numero espositori",
```

```
temp.nome as "Categoria"
```

```
From temp
```

```
group by nome
```

```
order by count(DISTINCT id_utente) DESC
```

Numero espositori	Categoria
3	Scultura
1	Installazione artistica
1	Mosaico
1	Pittura

## **Il progetto di massima della struttura funzionale dell'applicazione Web, realizzando, con appropriati linguaggi a scelta sia lato client che lato server, la parte che consente la visualizzazione dei dati di un contributo nella sezione ad accesso libero**

La traccia richiede di realizzare un'applicazione web che gestisce una manifestazione pubblica in cui un certo numero di espositori presenta i propri "contributi"

L'applicazione prevede tre momenti:

- 1) gli aspiranti espositori si registrano con username e password e presentano la propria candidatura ad esporre uno o più contributi
- 2) il personale dell'Associazione analizza le schede di candidatura, sceglie i contributi che saranno esposti nella manifestazione e assegna un'area a ciascun espositore
- 3) ad accesso libero, i contributi dettagliati della manifestazione, degli espositori e dell'area di esposizione vengono presentati online

### **Pagina di visualizzazione dei contributi ad accesso libero**

Il codice è composto dal file **seleziona.php** e dal file **visualizza.php**

La pagina **seleziona.php** visualizza un menù a discesa con i contributi esposti nella manifestazione. Selezionando un contributo viene richiamata la pagina **visualizza.php** che consente di visualizzare tutte le informazioni disponibili riguardanti il contributo e l'espositore

Il codice proposto è molto essenziale e non viene utilizzato alcun foglio di stile.

### **Codice "seleziona.php"**

```
<?php
if(isset($_POST['contributo']))header("Location: visualizza.php?contributo=".$_POST['contributo']);
// richiama la pagina visualizza.php passando il parametro contributo=id relativo al contributo selezionato
?>
<!DOCTYPE html>
<html>
<head>
<title>Visualizzazione contributi</title><meta charset="utf-8">
</head>
<body>
<p>Associazione culturale X - Manifestazione culturale Y</p>
<h2>Elenco dei contributi</h2>
<form method="POST" enctype="multipart/form-data">
<select name="contributo" onchange="this.form.submit()" >
<?php
$conn=mysqli_connect("localhost","my_username","my_password","contributi"); //si connette al database
if($conn) {
    mysqli_query($conn,"SET NAMES 'utf8'");
    mysqli_set_charset($conn,'utf8');
    // per accettare correttamente le lettere accentate della tabella
}
else {
    echo "Errore numero ".mysqli_errno($conn)."<br/>Descrizione errore: "
        .mysqli_error($conn);
}
}
```

```
// Legge le righe della tabella candidature accettate
$s = "Select candidature.id,titolo,utenti.cognome,utenti.nome
      From candidature,utenti
      Where accettazione='A'
      And utenti.id=candidature.id_utente
      Order By titolo";
$query=mysqli_query($conn,$s);
// popola la select con i dati della tabella
echo '<option value="">Seleziona il contributo </option>';
while($r = mysqli_fetch_array($query)) {
    $v=$r[0];           // id contributo
    $t=$r[1];           // titolo del contributo
    $n=$r[2]." ".$r[3]." - "; //cognome e nome espositore
    if(strlen($t)>60) $t=substr($t,0,59);
    echo "<option value='$v'>$n.$t</option>";
}
mysqli_close($conn);
?>
</select>
</form>
<br/><br/>
</body>
</html>
```

### **Codice "visualizza.php"**

```
<?php
    $id_contributo=(int)$_GET['contributo'];
    //recupera l'id del contributo selezionato
?>
<!DOCTYPE html>
<html>
<head>
    <title>Visualizzazione contributi</title><meta charset="utf-8">
</head>
<body>
<p>Associazione culturale X - Manifestazione culturale Y</p>
<?php
    $conn=mysqli_connect("localhost"," my_username ","my_password", "contributi"); //si connette al database
    if($conn) {
        mysqli_query($conn,"SET NAMES 'utf8'");
        mysqli_set_charset($conn,'utf8');
        // per accettare correttamente le lettere accentate dalle tabelle
    }
    else{
        echo "Errore numero ".mysqli_errno($conn)."<br/>Descrizione errore: " .mysqli_error($conn);
    }
    // legge i dati relativi contributo che si vuole visualizzare
    $s = "Select data,titolo, sintesi,url_img,url_approfondimento,id_utente
          From candidature
          where id=$id_contributo";
    $query=mysqli_query($conn,$s);
    $r=mysqli_fetch_array($query);
    $d=$r[0];           //data
    $d=date_create($d);
```

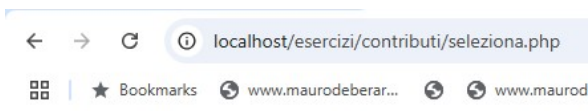
```
$data=date_format($d,"d/m/Y"); // trasformo la data nel formato d/m/Y
$titolo=$r[1]; //titolo
$sintesi=$r[2]; //sintesi
$img=$r[3]; //url immagine
$approfondimento=$r[4]; //url approfondimento
$utente=$r[5]; //id espositore
// legge i dati dell'espositore
$s = "Select cognome,nome, url_cve,id_area
      From utenti
      where id=$utente";
$query=mysqli_query($conn,$s);
$r=mysqli_fetch_array($query);
$name=$r[0]." ".$r[1]; //cognome e nome dell'espositore
$cve=$r[2]; //url curriculum vitae dell'espositore
$area=$r[3]; // area assegnata all'espositore
// legge la categoria associata al contributo
$s = "Select categorie.nome
      From candidature, associazioni,categorie
      where candidature.id=$id_contributo
      and candidature.id=associazioni.id_candidatura
      and associazioni.id_categoria=categorie.id";
$query=mysqli_query($conn,$s);
$r=mysqli_fetch_array($query);
$categoria=$r[0]; //categoria di classificazione associata al contributo
// legge il nome dell'area della manifestazione assegnata all'espositore
if($area==0) {
    $area= "Non ancora assegnata";
}
else{
    $s = "Select aree.nome
          From aree
          where id=$area";
    $query=mysqli_query($conn,$s);
    $r=mysqli_fetch_array($query);
    $area=$r[0]; //area associata all'espositore
}
mysqli_close($conn);
echo"<h2><font color=#00008B>Contributo Id=$id_contributo</font></h2>";
echo"<b>Informazioni sull'espositore</b>";
echo"<ul><li>.$nome.</li>";
echo"<li>Area della manifestazione: ".$area."</li>";
if($cve==""){
    echo "<li>Curriculum vitae europeo non disponibile";
}
else{
    echo"<li><a href='$cve' target='_blank'>Curriculum vitae europeo</a>";
}
echo"</li></ul>";
echo"<b>Informazioni sul contributo</b>";
echo"<ul><li>Titolo<br/><textarea readonly='readonly' rows='3' cols='80'>.$titolo.</textarea>";
echo"</li><li>Data di presentazione candidatura: ".$data;
echo"</li><li>Categoria: ".$categoria;
```

```
echo"</li><li>Sintesi<br/><textarea readonly='readonly' rows='10' cols='80'> .$.sintesi."</textarea></li>";
if($img==""){
    echo "<li>Immagine non disponibile";
}
else{
    echo"<li>Immagine allegata (Cliccare sopra per ingrandirla)<br/><a href='$img' target='_blank'>
    <img src='$img' style='max-width:100px;max-height:100px'></a>";
}
echo"</li>";
if($approfondimento==""){
    echo "<li>Pagine di approfondimento non disponibili";
}
else{
    echo"<li><a href='$approfondimento' target='_blank'>Pagine di approfondimento</a>";
}
echo"</li></ul>";
?>
<br/>
<a href="selezione.php"><button style="margin-left:200px">Torna alla pagina precedente</button></a>
</body>
</html>
```

## Test del codice

Per il test del codice ho utilizzato Wamp Server 3.3.0 64bit, PHP 8.0.26, MySql 8.0.31 e PHpMyAdmin 5.2.0.

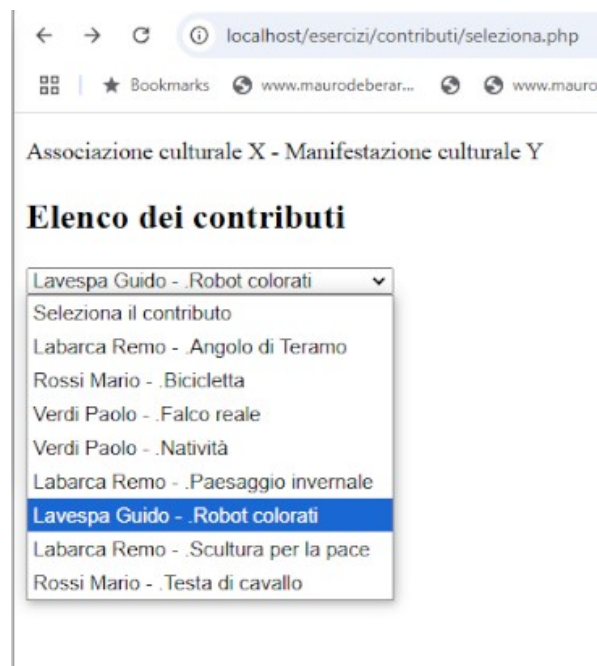
### selezione.php



Associazione culturale X - Manifestazione culturale Y

## Elenco dei contributi

Seleziona il contributo



[visualizza.php](#)



Associazione culturale X - Manifestazione culturale Y

## Contributo Id=3

### Informazioni sull'espositore

- Lavespa Guido
- Area della manifestazione: Salone delle feste
- Curriculum vitae europeo non disponibile

### Informazioni sul contributo

- Titolo

Robot colorati

- Data di presentazione candidatura: 03/02/2025
- Categoria: Installazione artistica
- Sintesi

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. "Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui

- Immagine allegata (Cliccare sopra per ingrandirla)



- [Pagine di approfondimento](#)

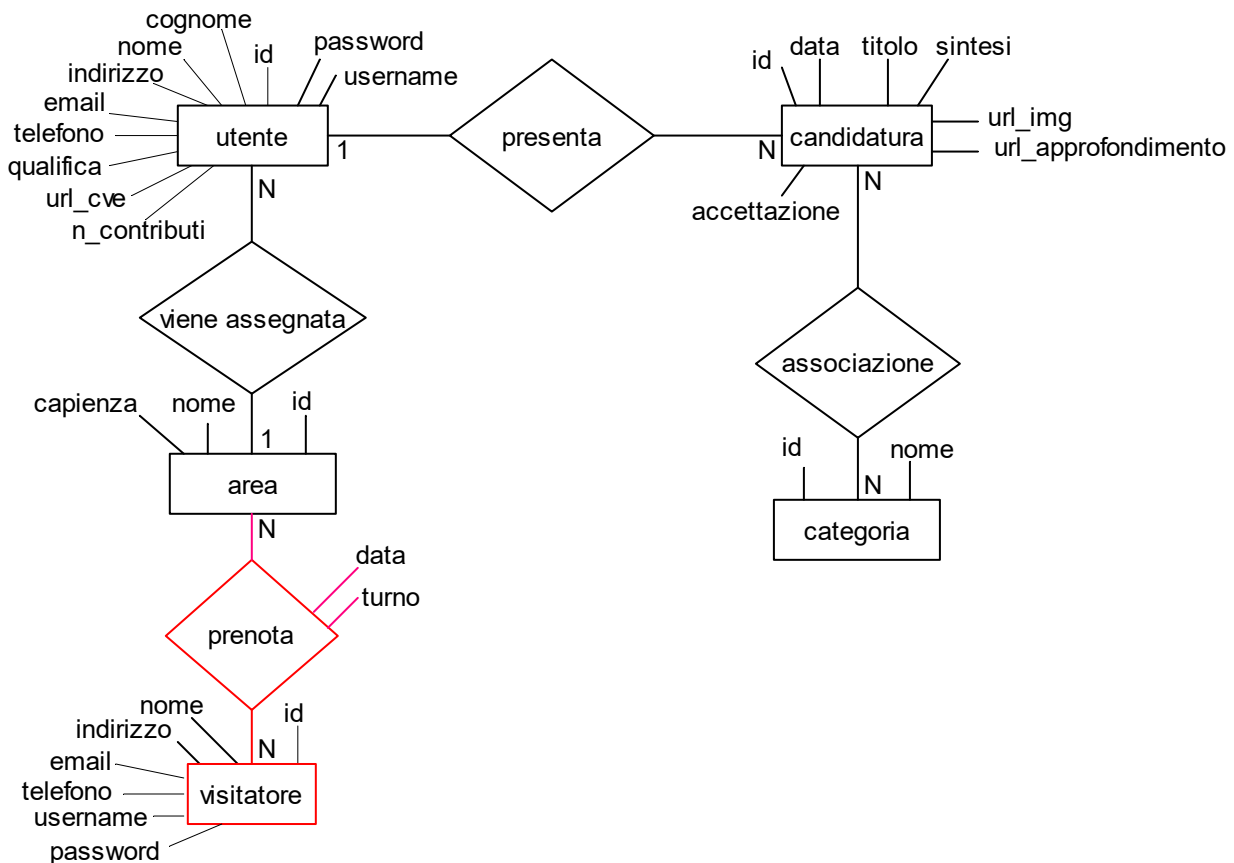
[Torna alla pagina precedente](#)

## Soluzione seconda parte

- I. In relazione al tema sviluppato nella prima parte, e considerando che ogni area ha una propria capienza massima che non può essere superata per motivi di sicurezza, l'accesso dei visitatori alle diverse aree viene organizzato in turni. Il candidato integri la base di dati (schema concettuale e logico) in modo da gestire on-line la prenotazione dei visitatori. Ogni visitatore registra un proprio profilo e poi può prenotare una o più aree e turni per la visita. Il sistema di registrazione impedirà la scelta di un'area e turno per i quali è già stata raggiunta la capienza limite. Il candidato realizzi, con appropriati linguaggi a scelta sia lato client che lato server, la parte dell'applicazione Web che consente ad un visitatore registrato di svolgere le operazioni di prenotazione.

Integrazione della base di dati (integrazione evidenziata in rosso)

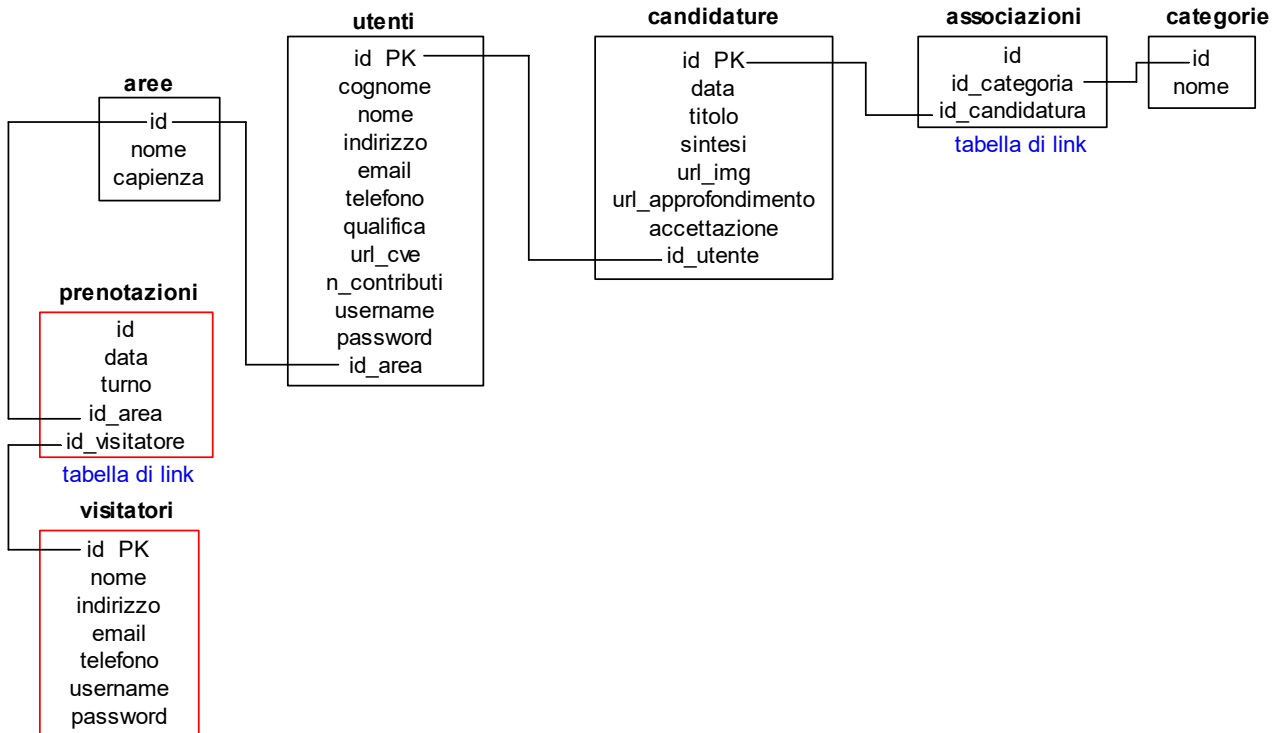
### Schema concettuale (Diagramma E/R)



### Letture dell'integrazione del diagramma E/R

- Ogni visitatore registrato prenota la visita di N aree con data e turno, ogni area è prenotata da N visitatori

## Schema logico



## Definizione delle relazioni in linguaggio SQL

### Struttura delle tabelle visitatori e prenotazioni

tabella	campi	key	tipo	lung.	descrizione/note
visitatori	id	PK	Int		Contatore auto_increment
	nome		VarChar	40	
	indirizzo		VarChar	80	
	email		VarChar	30	
	telefono		VarChar	30	
	username		VarChar	20	
	password		VarChar	32	Si utilizza la codifica MD5()
prenotazioni tabella di link	id	PK	Int		Contatore auto_increment
	data		date		
	turno	Int			Esempio: turno 1
	id_area	Int			Fa riferimento ad aree.id
	id_visitatore	Int			Fa riferimento a visitatori.id

## Query MySql per la creazione delle tabelle visitatori e prenotazioni (DB contributi)

```
CREATE TABLE IF NOT EXISTS visitatori(  
id INT AUTO_INCREMENT PRIMARY KEY,  
nome VARCHAR(40),  
indirizzo VARCHAR(80),  
email VARCHAR(30),  
telefono VARCHAR(30),  
username VARCHAR(20),  
password VARCHAR(32));
```

```
CREATE TABLE IF NOT EXISTS prenotazioni (  
id INT AUTO_INCREMENT PRIMARY KEY,  
data DATE,  
turno INT,  
id_area INT,  
id_visitatore INT,  
FOREIGN KEY (id_area) REFERENCES aree(id),  
FOREIGN KEY (id_visitatore) REFERENCES visitatori(id));
```

### Prenotazione visite

Il codice è composto dal file **prenota.php** (lato client) e dal file **registraprenotazioni.php** (lato server)

La pagina prenota.php presenta un form che consente al visitatore di prenotare la visita a una o più aree espositive. Poiché il sistema impedisce "la scelta di un'area e turno per i quali è già stata raggiunta la capienza limite", si svolge un "dialogo" tra visitatore e sistema:

- il visitatore inserisce nel form area, data e turno della prenotazione che vuole effettuare
- il sistema controlla che la visita sia possibile. In caso positivo registra la prenotazione, viceversa invita il visitatore a prenotare in altra data e/o in altro turno

Per rendere le pagine interattive ed evitare di ricaricare più volte la pagina prenota.php , utilizziamo AJAX .

### Ipotesi di lavoro

1. la pagina è accessibile solo agli utenti registrati e autenticati
2. l'id del visitatore viene recuperato dalla variabile di sessione \$\_SESSION('id') impostata nella fase di login
3. l'accesso ai visitatori è organizzato nei seguenti turni di 2 ore ciascuno: Turno1 9:11, Turno2 11:13, Turno3 15:17, Turno4 17:19
4. le date previste per le visite sono inserite dagli organizzatori in una tabella "datevisite" (id,data)

La soluzione proposta è molto essenziale e didattica, ma gli studenti più motivati possono integrare e migliorare il codice:

- a) curando l'aspetto estetico attraverso i fogli di stile
- b) prevedendo la possibilità per il visitatore di visualizzare le prenotazioni già effettuate

## prenota.php

```
<?php
    session_start();
    //recupero l'id del visitatore
    $id_visitatore=$_SESSION['id'];
?>
<!DOCTYPE html>
<html>
<head>
    <title>Prenotazione visite</title><meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <script src="http://code.jquery.com/jquery-latest.js"></script>
    <script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
</head>
<script type="text/javascript">
$(document).ready(function()
{
    $('#reset').click(function(){
        // il reset pulisce anche il div messaggio e abilita il tasto "prenota"
        // se è stato disabilitato
        $("#messaggio").empty();
        $("#submit").removeAttr('disabled');
    });
    $('#submit').click(function()
    {
        var ar=$("#area:checked").val();
        var dv=$("#datavisita").val();
        var tu=$("#turno:checked").val();
        var vis=$("#visitatore").val();
        if(isNaN(ar) || dv=="" || isNaN(tu)){
            alert("Dati incompleti: selezionare area, data e turno");
            return false;
        }
        // lo script ajax consente l'esecuzione in background dello script registraprenotazioni.php
        $.ajax({
            url:"registraprenotazioni.php",
            method:"POST",
            data:{area:ar,datavisita:dv,turno:tu,visitatore:vis},
            success: function (risposta)
            {
                //visualizza la risposta dello script registraprenotazioni.php
                $('#messaggio').html(risposta);
                // disabilita il tasto "Prenota" per evitare che si possa effettuare più volte la stessa prenotazione
                $("#submit").prop("disabled", true);
            }
        });
    });
    return false;
});
});
});
</script>
```

```
<body>
<p>Associazione culturale X - Manifestazione culturale Y</p>
<h2>Prenotazione visite</h2>
<?php
$conn=mysqli_connect("localhost","root","","contributi"); //si connette al database
if($conn) {
    mysqli_query($conn,"SET NAMES 'utf8'");
    mysqli_set_charset($conn,'utf8');
    // per accettare correttamente le lettere accentate dalle tabelle
}
else{
    echo "Errore numero ".mysqli_errno($conn)."<br/>Descrizione errore: " .mysqli_error($conn);
}
// legge i dati del visitatore
$s = "Select nome,indirizzo
      From visitatori
      where id=$id_visitatore";
$query=mysqli_query($conn,$s);
$r=mysqli_fetch_array($query);
$nomevisitatore=$r[0];           // nome visitatore
$indirizzo=$r[1];                // indirizzo visitatore
echo "<p><font color='#0000F0'>$nomevisitatore<br/>$indirizzo</font></p>";
// form *****
echo "<form method='POST' enctype='multipart/form-data'>";
echo "<input type='text' id='visitatore' name='visitatore' value=$id_visitatore style='display:none'>";
$conn=mysqli_connect("localhost","my_username","my_password","contributi"); //si connette al database
if($conn) {
    mysqli_query($conn,"SET NAMES 'utf8'");
    mysqli_set_charset($conn,'utf8');// per accettare correttamente le lettere accentate della tabella
}
else {
    echo "Errore numero ".mysqli_errno($conn)."<br/>Descrizione errore: " .mysqli_error($conn);
}

// Legge le righe della tabella aree
$s = "Select id,nome from aree";
$query=mysqli_query($conn,$s);
// popola l'elenco non numerato con i dati della tabella
echo "<p><b>Seleziona l'area espositiva</b>";
echo "<ul>";
$idarea=0;
while($r = mysqli_fetch_array($query)) {
    $idarea=$r[0];           // id area
    $nomearea=$r[1]; // nome area
    echo "<li><input type='radio' name='area' id='area' value=$idarea>$nomearea</li>";
}
echo "</ul></p>";
```

```
echo "<p><b>Seleziona la data</b> </p>";
// Legge le righe della tabella datevisite
$s = "Select id,data from datevisite order by data";
$query=mysqli_query($conn,$s);
// popola la select con i dati della tabella
echo "<select name='datavisita' id='datavisita' style='width:200px'>";
echo "<option value=''></option>";
while($r = mysqli_fetch_array($query)) {
    $iddata=$r[0]; // id data
    $datavisita=$r[1]; // data
    $d=date_create($datavisita);
    $data_dmy=date_format($d,"d/m/Y");
    echo "<option value='$datavisita'>$data_dmy</option>";
}
echo "</select>";
mysqli_close($conn);
?>
<p><b>Seleziona il turno </b>
<ul>
<li><input type='radio' name='turno' id='turno' value=1>Turno1 09:11</li>
<li><input type='radio' name='turno' id='turno' value=2>Turno2 11:13</li>
<li><input type='radio' name='turno' id='turno' value=3>Turno3 15:17</li>
<li><input type='radio' name='turno' id='turno' value=4>Turno4 17:19</li>
</ul></p> <br/>
<button type="submit" id="submit" style="width:62px">Prenota</button>
<button type="reset" id="reset" style="width:62px">Reset</button><br/><br/>
<span id="messaggio"></span><br/><br/>
</form>
</body>
</html>
```

## registraprenotazioni.php

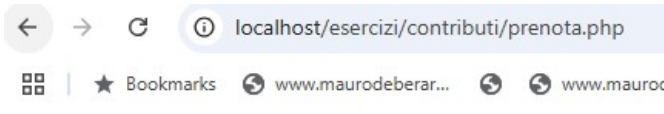
```
<?php
if(isset($_POST["area"]))$area=$_POST["area"];
if(isset($_POST["datavisita"]))$datavisita=$_POST["datavisita"];
if(isset($_POST["turno"]))$turno=$_POST["turno"];
if(isset($_POST["visitatore"]))$visitatore=$_POST["visitatore"];
$conn=mysqli_connect("localhost","my_username","my_password","contributi");
// legge la capienza dell'area
$s="select capienza,nome from aree where id=$area";
$q=mysqli_query($conn,$s);
$r=mysqli_fetch_array($q);
$capienza=$r[0];
$nomearea=$r[1];
// conta i visitatori che nella data e nel turno hanno prenotato l'area
$s="select count(*) from prenotazioni
    where data='$datavisita'
    and turno=$turno
    and id_area =$area";
$q=mysqli_query($conn,$s);
$r=mysqli_fetch_array($q);
$nprenotazioni=$r[0]; // numero di prenotazioni
$d=date_create($datavisita);
$data_dmy=date_format($d,"d/m/Y");
```

```
if($nprenotazioni<$capienza){  
  // se il numero delle prenotazioni non ha raggiunto la capienza  
  // la prenotazione è possibile e viene registrata  
  $s="Insert into prenotazioni  
    (data,turno,id_area,id_visitatore)  
    Values('$datavisita',$turno,$area,$visitatore)";  
  $q=mysqli_query($conn,$s);  
  echo "Data e turno:&nbsp;$data_dmy &nbsp;&nbsp;&nbsp;$turno<br/>  
    Area: $nomearea <br/>  
    <span style='color:#0000FF'>Prenotazione acquisita correttamente<br/>  
    E' stata inviata una conferma al suo indirizzo email</span><br/>";  
}  
else{  
  echo "Data e turno:&nbsp;$data_dmy &nbsp;&nbsp;&nbsp;$turno<br/>  
    Area: $nomearea <br/>  
    <span style='color:#FF0000'>Prenotazione impossibile, raggiunta la capienza limite</span><br/>";  
}  
mysqli_close($conn);
```

?>

## Test del codice

Per il test del codice ho utilizzato Wamp Server 3.3.0 64bit, PHP 8.0.26, MySql 8.0.31 e PHpMyAdmin 5.2.0



Associazione culturale X - Manifestazione culturale Y

## Prenotazione visite

Ugo Foscolo  
Largo del Tordino snc Teramo

### Seleziona l'area espositiva

- Salone delle feste
- Salone reale
- Corridoio
- Scuderia

### Seleziona la data

25/02/2025

### Seleziona il turno

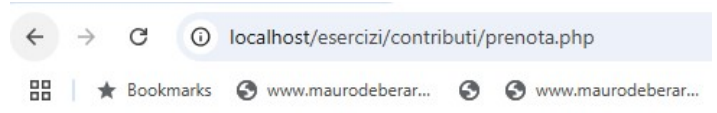
- Turno1 09:11
- Turno2 11:13
- Turno3 15:17
- Turno4 17:19

Prenota Reset

Data e turno: 25/02/2025 1

Area: Salone delle feste

Prenotazione impossibile, raggiunta la capienza limite



Associazione culturale X - Manifestazione culturale Y

## Prenotazione visite

Ugo Foscolo  
Largo del Tordino snc Teramo

### Seleziona l'area espositiva

- Salone delle feste
- Salone reale
- Corridoio
- Scuderia

### Seleziona la data

25/02/2025

### Seleziona il turno

- Turno1 09:11
- Turno2 11:13
- Turno3 15:17
- Turno4 17:19

Prenota Reset

Data e turno: 25/02/2025 2

Area: Salone delle feste





Prenotazione acquisita correttamente

E' stata inviata una conferma al suo indirizzo email



- II. In relazione al tema sviluppato nella prima parte, si consideri che ogni utente registrato ha la possibilità di creare, rivedere, modificare, cancellare le proprie schede di candidatura fino ad una data limite prefissata, dopo di che ne potrà solo prendere visione. Il candidato produca il layout grafico della porzione dell'applicazione Web che consente agli utenti registrati di svolgere le operazioni specificate. Descriva poi la struttura di una guida on-line a supporto degli utenti registrati per illustrare tali funzionalità, scrivendo anche un esempio significativo dei contenuti.


L'utente registrato e "loggato" seleziona, tramite un menù a tendina, l'opzione "Le mie candidature". Viene visualizzata una tabella che presenta Id, Data e Titolo delle schede di candidatura già presentate e consente entro la data limite prefissata di inserire una nuova candidatura

### Le mie candidature

	Id	Data	Titolo
	1	15/01/2025	Paesaggio invernale
	2	23/01/2025	Angolo di Teramo
	8	05/02/2025	Scultura per la pace

Se la data limite prefissata dall'Associazione culturale che ha organizzato la manifestazione non è stata superata:

- Cliccando sull'icona  l'utente può creare una nuova scheda di candidatura attraverso un form di inserimento. In questa fase viene inserita una riga nella tabella MySQL "candidature"
- Cliccando sull'icona  della riga desiderata, si apre il form con i dati della candidatura già caricati dall'utente e recuperati dalla tabella MySQL "candidature". L'utente ha la possibilità modificare oppure cancellare la scheda di candidatura

Se la data limite è stata superata, l'utente può solo prendere visione delle schede di candidatura selezionandole tramite l'icona 

Riguardo alla guida online, una struttura a supporto degli utenti registrati nelle operazioni di creazione, visualizzazione, modifica e cancellazione di una candidatura, potrebbe essere la seguente:

- Creazione/Modifica/Cancellazione Schede
  - Eseguire il Login
  - Dettagli sui campi del form di inserimento/modifica
  - Creare una scheda
  - Caricare gli allegati
  - Visualizzare/Modificare una scheda
  - Cancellare una scheda

Per realizzare una guida online di supporto, occorre creare una tabella MySQL "guidaonline" del DataBase "contributi" che consenta di memorizzare i contenuti della guida.

### Struttura della tabella e Query MySQL per la creazione

tabella	campi	key	tipo	lung.	descrizione/note
guidaonline	id	PK	Int		Contatore auto_increment
	titolo		VarChar	60	
	contenuto		Text		

```
CREATE TABLE IF NOT EXISTS guidaonline(  
id INT AUTO_INCREMENT PRIMARY KEY,  
titolo VARCHAR(60),  
contenuto TEXT);
```

## Esempio di contenuto

Titolo	Contenuto
Cancellare una scheda	La cancellazione di una scheda di candidatura è possibile solo se la data limite prefissata dall'Associazione non è stata superata. In caso contrario: 1. esegui la procedura di login e dal menù utente dell'area riservata scegli "Le mie candidature". Viene visualizzata una tabella che consente di inserire una nuova candidatura oppure di gestire le schede di candidatura già presentate 2. seleziona la candidatura che intendi cancellare. Si apre il form con i dati della candidatura già caricati dall'utente 3. clicca sul pulsante "Cancella scheda". Viene visualizzata una finestra di conferma 4. clicca sul pulsante "Conferma cancellazione". La scheda viene eliminata in modo definitivo

- III. Il candidato presenti una panoramica delle tecnologie di sua conoscenza per lo sviluppo di applicazioni Web dinamiche. Approfondisca le differenze tra tali tecnologie, nelle funzionalità ed utilizzi tipici, e le esemplifichi, facendo riferimento al seguente caso concreto: in una applicazione Web viene richiesto di compilare tre campi di residenza con Regione, Provincia e Comune, che vengono impostati tramite tre listbox in cui la seconda e la terza presentano valori diversificati in funzione di quanto selezionato rispettivamente nella prima e nella seconda (es: se come regione viene selezionata Liguria dall'elenco delle Regioni, la listbox Provincia dovrà mostrare i valori Imperia, Savona, Genova, La Spezia).

In ambito didattico, per lo sviluppo di applicazioni Web dinamiche sono molto utilizzate le seguenti tecnologie:

- Lato client (Frontend): HTML5, CSS3, Javascript. Per rendere le pagine interattive e professionali si utilizza anche AJAX che consente di aggiornare parti di una pagina web senza doverla ricaricare
- Lato server (Backend): PHP
- Database (memorizza e gestisce i dati dell'applicazione web): MySql

Le tecnologie di Frontend consentono all'utente di interagire con l'applicazione mentre quelle di Backend consentono di connettersi al database per leggere e scrivere i dati, elaborare le richieste degli utenti e restituire le risposte.

Una soluzione concreta per la selezione dinamica di Regioni, Province e Comuni (recuperata dalle lezioni di Informatica nella classe VBI dell'ITIS di Teramo dell'anno scolastico 2012/2013) l'ho già proposta in questo sito nell'ormai lontano novembre 2020 al seguente link:

[https://www.maurodeberardis.it/index.php?option=com\\_content&view=article&id=87:come-gestire-select-dinamiche-con-jquery-e-ajax&catid=12&Itemid=328](https://www.maurodeberardis.it/index.php?option=com_content&view=article&id=87:come-gestire-select-dinamiche-con-jquery-e-ajax&catid=12&Itemid=328)

Consultare le pagine del link può essere molto utile per quanti vogliono approfondire e comprendere meglio il codice proposto.

Nella pagina "Scarica la soluzione PHP" uno script SQL permette di importare le tabelle MySql regioni, province e comuni (le tabelle non sono aggiornate)

### Come gestire select dinamiche con JQuery e Ajax

© Pubblicato Mercoledì, 25 Novembre 2020 15:49

👤 Visite: 5591

#### Select dinamiche Regioni-Province-Comuni

Seleziona la Regione ▼

Seleziona la Provincia ▼

Seleziona il Comune ▼

Un tutorial e una soluzione PHP per gestire le tre select Regioni, Province e Comuni in cascata

Il problema delle select dinamiche è stato già stato proposto nell'Area Download un paio di anni fa con una soluzione, limitata alle select di province e comuni, che utilizzava Php e JavaScript e prevedeva il ricaricamento della pagina quando la select dei comuni veniva filtrata in base alla scelta della provincia. Riprendo il problema e propongo una soluzione in cui, grazie a jQuery e Ajax, le tabelle MySql popolano le tre select in modo dinamico evitando il refresh della pagina.

[Prova il codice](#)

[Scarica il tutorial](#)

[Scarica la soluzione PHP](#)

Con piccolissime variazioni, ripropongo di seguito quella stessa soluzione.

## Funzionamento

### Select dinamiche Regioni-Province-Comuni

Seleziona la Regione ▼

Seleziona la Provincia ▼

Seleziona il Comune ▼

### Select dinamiche Regioni-Province-Comuni

Seleziona la Regione ▼

Seleziona la Regione

**ABRUZZO**

BASILICATA

CALABRIA

CAMPANIA

EMILIA-ROMAGNA

FRIULI-VENEZIA GIULIA

LAZIO

LIGURIA

LOMBARDIA

MARCHE

MOLISE

PIEMONTE

PUGLIA

SARDEGNA

SICILIA

TOSCANA

TRENTINO-ALTO ADIGE

UMBRIA

VALLE D'AOSTA/VALLE D'AOSTE

### Select dinamiche Regioni-Province-Comuni

ABRUZZO ▼

Seleziona la Provincia ▼

Seleziona la Provincia

Chieti

L'Aquila

Pescara

**Teramo**

### Select dinamiche Regioni-Province-Comuni

ABRUZZO ▼

Teramo ▼

Seleziona il Comune ▼

Seleziona il Comune

Alba Adriatica

Ancarano

Arsita

Atri

Basciano

Bellante

Bisenti

Campoli

Canzano

Castel Castagna

Castellalto

Castelli

Castiglione Messer Raimondo

Castilenti

Cellino Attanasio

Cermignano

Civitella del Tronto

**Colledara**

Colonnella

### Select dinamiche Regioni-Province-Comuni

ABRUZZO ▼

Teramo ▼




Colledara ▼

Scegliendo la regione nella prima select, la seconda select viene popolata con le province della regione selezionata. Scegliendo la provincia, la terza select viene popolata con i comuni della provincia selezionata.

L'applicazione è basata su tre select dinamiche che vengono popolate da altrettante tabelle MySQL; grazie all'utilizzo di jQuery e Ajax si evita il refresh della pagina.

È una classica applicazione client-server in cui lato server viene eseguito uno script Php che si interfaccia con un DataBase MySQL.

Le tabelle MySQL regioni, province e comuni che popolano le tre select e alle quali il codice fa riferimento hanno le seguenti strutture:

regioni		province			comuni		
#	Nome	#	Nome	Tipo	#	Nome	Tipo
1	id 	1	id 	int	1	id 	int
2	ripartizione_geografica	2	codice_regione	int	2	codice_provincia	int
3	codice	3	codice	int	3	codice	int
4	nome	4	nome	tinytext	4	nome	tinytext
		5	sigla	tinytext			

La soluzione è composta di due files: selectdinamiche.php e filtra.php

### selectdinamiche.php

```
<!DOCTYPE HTML>
<html>
<!--
/*
* Nel tutorial "Select dinamiche di regioni, province e comuni con Php, JQuery e Ajax"
* che si può scaricare nell'Area Download - Tutorial e approfondimenti del sito www.maurodeberardis.it
* questo codice è corredato da commenti e spiegazioni
*/
-->
<head>
  <title>Select dinamiche MDB2020</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
  <script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
  <script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
</head>
<style>
select{
  width:300px;
  height:30px;
  font:18px;
  padding:5px;
}
</style>
```

```
<?php
$server="localhost";
$username="my_username";
$password="my_password";
$database="tabelle";
$conn=mysqli_connect($server,$username,$password,$database);
if($conn){
    mysqli_query($conn,"SET NAMES 'utf8'");
    mysqli_set_charset($conn,'utf8');
}
else{
    echo "Errore numero ".mysqli_errno($conn). "nella connessione
    al server".<br/>Descrizione errore ".mysqli_error($conn);
}
$s = "Select codice,nome From regioni Order By nome" ;
$query = mysqli_query($conn,$s);
?>
<body>
<div style="text-align:center">
<h4>Select dinamiche Regioni-Province-Comuni</h4>
<select id = "regione" >
<option value = "0" >Seleziona la Regione </option >
<?php
    while ($r=mysqli_fetch_array($query)){
        echo '<option value = "' . $r[0]. "'> ' . $r[1]."</option>";
    }
?>
</select ><br/><br/>
<select id = "provincia" >
    <option value="0">Seleziona la Provincia</option >
</select><br/><br/>
<select id = "comune" >
    <option value="0">Seleziona il Comune </option >
</select >
</div>
<script>
$(document).ready(function(){
    $('#regione').on('change', function(){
        codRegione = $(this).val();
        if(codRegione!=0){
            $.ajax({
                type:'POST',
                url:'filtra.php',
                data:'codRegione='+codRegione,
                success:function(rispostahtml){
                    $('#provincia').html(rispostahtml);
                    $('#comune').html('<option value="">Seleziona la Provincia</option>');
                }
            });
        }
        else{
            $('#provincia').html('<option value="">Seleziona la Provincia</option>');
            $('#comune').html('<option value="">Seleziona il Comune </option>');
        }
    });
});
});
```

```
$('#provincia').on('change', function(){
  codProvincia = $(this).val();
  if(codProvincia!=0){
    $.ajax({
      type:'POST',
      url:'filtra.php',
      data:'codProvincia='+codProvincia,
      success:function(rispostahtml){
        $('#comune').html(rispostahtml);
      }
    });
  }else{
    $('#comune').html('<option value="">Seleziona il Comune</option>');
  }
});
</script>
</body>
</html>
```

### filtra.php

```
<?php
header("Content-type: text/html; charset=utf-8");
$server="localhost";
$username="root";
$password="";
$databse="tabelle";
$conn=mysqli_connect($server,$username,$password,$databse);
if(!empty($_POST["codRegione"])){
  $codRegione=$_POST["codRegione"];
  $s = "Select codice,nome From province Where codice_regione = ".$codRegione." Order By nome";
  $query=mysqli_query($conn,$s);
  echo '<option value="">Seleziona la Provincia</option>';
  while($r = mysqli_fetch_array($query)){
    echo '<option value="'.$r[0]."'>'.$r[1].'</option>';
  }
}
else{
  if(!empty($_POST["codProvincia"])){
    $codProvincia=$_POST["codProvincia"];
    $s = "Select id,nome From comuni Where codice_provincia = ".$codProvincia." Order By nome";
    $query = mysqli_query($conn,$s);
    echo '<option value="">Seleziona il Comune</option>';
    while($r = mysqli_fetch_array($query)){
      echo '<option value="'.$r[0]."'>'.$r[1].'</option>';
    }
  }
}
?>
```

**IV. Il candidato illustri le attività tipiche di un amministratore di una base di dati, corredandole con esempi significativi di comandi SQL specifici per lo svolgimento delle funzioni a lui riservate.**

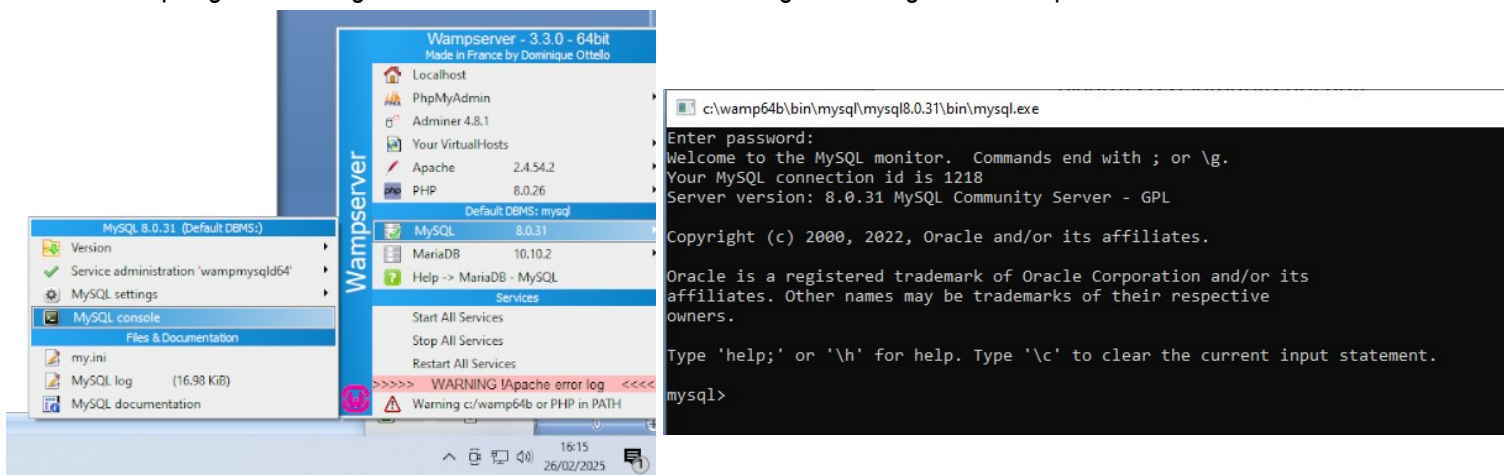
L'amministratore della base di dati (DBA -Database Administrator) è il responsabile della gestione, della manutenzione e della sicurezza dei database aziendali

Attività più importanti

- Installazione, configurazione e aggiornamento del DBMS (Database Management System ovvero Sistema di gestione del Database)
- Monitoraggio del database e risoluzione di problemi, ottimizzazione delle prestazioni
- Applicazione di rigorose procedure di backup e ripristino dei dati
- Implementazione di misure di sicurezza preventive (autenticazione, firewall, crittografia) a difesa della sicurezza e integrità dei dati
- Aggiornamento del software
- Creazione degli utenti e gestione dei permessi, indicando chi può accedere alle varie tabelle del database e con quali privilegi

Il DBA conosce in modo approfondito il DBMS utilizzato (in questa soluzione MySql) e padroneggia i comandi SQL specifici per lo svolgimento delle attività sopra elencate.

Utilizzando WampServer 3.3.0 entriamo nella console di MySql ed eseguiamo i comandi SQL relativi a tre esempi significativi di gestione dell'attività di Creazione degli utenti e gestione dei permessi



**1. Creare un utente**

CREATE USER 'username'@'host' IDENTIFIED BY 'password';

```
mysql> CREATE USER 'pippo'@'localhost' IDENTIFIED BY 'pluto';  
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> CREATE USER 'programmatore1'@'localhost' IDENTIFIED BY 'XYZ2025$';  
Query OK, 0 rows affected (0.01 sec)
```

**2. Assegnare ad un utente tutti i privilegi su un database specifico**

GRANT tipo\_privilegio ON database.\* TO 'username'@'localhost';

```
mysql> GRANT ALL PRIVILEGES ON contributi.* TO 'pippo'@'localhost';  
Query OK, 0 rows affected (0.00 sec)
```

**3. Assegnare ad un utente il privilegio di sola lettura di tutti i databases e tutte le tabelle**

GRANT SELECT ON \*.\* TO 'username'@'localhost';

```
mysql> GRANT SELECT ON *.* TO 'programmatore1'@'localhost';  
Query OK, 0 rows affected (0.00 sec)
```