

## Ministero dell'istruzione e del merito

**A038 - ESAME DI STATO CONCLUSIVO DEL SECONDO CICLO DI ISTRUZIONE**  
**Indirizzo ITIA - INFORMATICA E TELECOMUNICAZIONI ARTICOLAZIONE "INFORMATICA"**

Sessione straordinaria 2023  
Seconda prova scritta **Disciplina: INFORMATICA**

***Il candidato svolga la prima parte della prova e due tra i quesiti proposti nella seconda parte.***

### **PRIMA PARTE**

Tra i maggiori paesi europei produttori di miele vi è l'Italia, che vanta oltre 60 diverse varietà di miele, molte delle quali del tutto uniche proprio in virtù delle diverse peculiarità locali e territoriali (dalle fioriture alpine a quelle della macchia mediterranea, dalle coste adriatiche alle isole). La produzione è gestita da circa 60 mila apicoltori.

Considerata l'importanza di tale patrimonio economico, naturalistico e anche culturale, le principali associazioni di settore hanno convenuto sulla necessità di realizzare una banca dati online che consenta la raccolta, gestione e consultazione dei dati relativi alla produzione dei mieli italiani.

Per poter operare una distinzione tra i vari mieli sono state individuate le seguenti tipologie:

- Mieli ad identità nazionale, ossia prodotti in molte o tutte le regioni italiane
- Mieli ad identità regionale, ossia presenti in una o più specifiche regioni
- Mieli ad identità territoriale, ossia mieli presenti in alcuni specifici territori
- Mieli D.O.P., ossia a denominazione di origine protetta

Ogni miele ha quindi una sua denominazione ed appartiene ad una certa tipologia (ad esempio il "Millefiori" è un miele ad identità nazionale, "Erica" è un miele ad identità regionale, "Asfodelo" ad identità territoriale, il "Miele delle Dolomiti Bellunesi" è un miele DOP). Il miele viene prodotto in "apiari": un apiario è costituito da una o più arnie (struttura costruita dall'uomo per l'allevamento delle api) tra loro vicine.

Il progetto prevede che ogni apicoltore fornisca i dati per ciascun miele prodotto annualmente attraverso i propri apiari.

Per tenere conto delle specificità territoriali e delle produzioni locali, ciascun apiario, che produce una certa quantità annuale di un unico miele, è individuato attraverso le seguenti informazioni: codice identificativo, numero di arnie che lo compongono, località, comune, provincia, regione.

La gestione delle operazioni sulla base di dati dovrà essere svolta via web attraverso un opportuno insieme di pagine dinamiche, di cui alcune ad accesso esclusivo degli apicoltori opportunamente registrati, i quali immetteranno i dati relativi alle loro produzioni annuali; altre pagine saranno di libero accesso e destinate al pubblico ai fini della consultazione dei dati.

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. la definizione in linguaggio SQL di un sottoinsieme delle relazioni della base di dati in cui siano presenti alcune di quelle che contengono vincoli di integrità referenziale e/o vincoli di dominio, se esistenti;
4. le interrogazioni espresse in linguaggio SQL che restituiscono:
  - a) l'elenco degli apicoltori che producono miele DOP in una determinata regione;
  - b) il numero complessivo di apiari per ciascuna regione;
  - c) le quantità di miele prodotto in Italia lo scorso anno per ciascuna delle quattro tipologie;
5. il progetto di massima della struttura dell'applicazione web per la gestione della realtà sopra presentata, distinguendo le funzionalità da implementare in base ai vari possibili profili di utenza;
6. una parte significativa dell'applicazione web che consente l'interazione con la base di dati, utilizzando appropriati linguaggi a scelta sia lato client che lato server.

## SECONDA PARTE

- I. In relazione al tema proposto nella prima parte si realizzi, in un linguaggio a scelta, una porzione di codice significativa della pagina dinamica che consente all'utente di ottenere l'elenco degli apicoltori di una certa regione, selezionando sulla pagina web la regione di interesse a partire da una rappresentazione in forma grafica del territorio italiano.
- II. In relazione al tema proposto nella prima parte, prevedendo eventuali integrazioni della base di dati, si realizzi in un linguaggio a scelta la pagina dinamica che visualizza l'elenco prodotto dalla query a) del punto 4 della prima parte, riportando in aggiunta per ciascun apicoltore l'eventuale link al suo sito web e la mail.
- III. In considerazione del fatto che il protocollo http è "stateless", il candidato descriva le possibili modalità di propagazione dei dati fra successive pagine web consultate in un sito, al fine di assicurarne la persistenza.
- IV. Si supponga che siano già state create le tabelle corrispondenti al seguente schema logico  
Progetti (id, titolo, budget, data\_inizio, data\_fine)  
Docenti (id, nome, cognome, posta\_elettronica, telefono)  
Si utilizzino opportuni comandi SQL per:
  - a) aggiungere nella tabella Progetti il campo che identifica il docente responsabile di ciascun progetto descritto nella tabella corrispondente
  - b) rinominare nella tabella Docenti il campo "posta\_elettronica" in "e-mail"

---

Durata massima della prova: 6 ore.

È consentito soltanto l'uso dei manuali di riferimento dei linguaggi di programmazione (language reference) e di calcolatrici scientifiche e/o grafiche purché non siano dotate di capacità di calcolo simbolico.

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 consegna della traccia.

## Soluzione prima parte

La traccia richiede di "realizzare una banca dati online che consenta la raccolta, gestione e consultazione dei dati relativi alla produzione dei mieli italiani" e suggerisce la realizzazione di un'applicazione web che consenta:

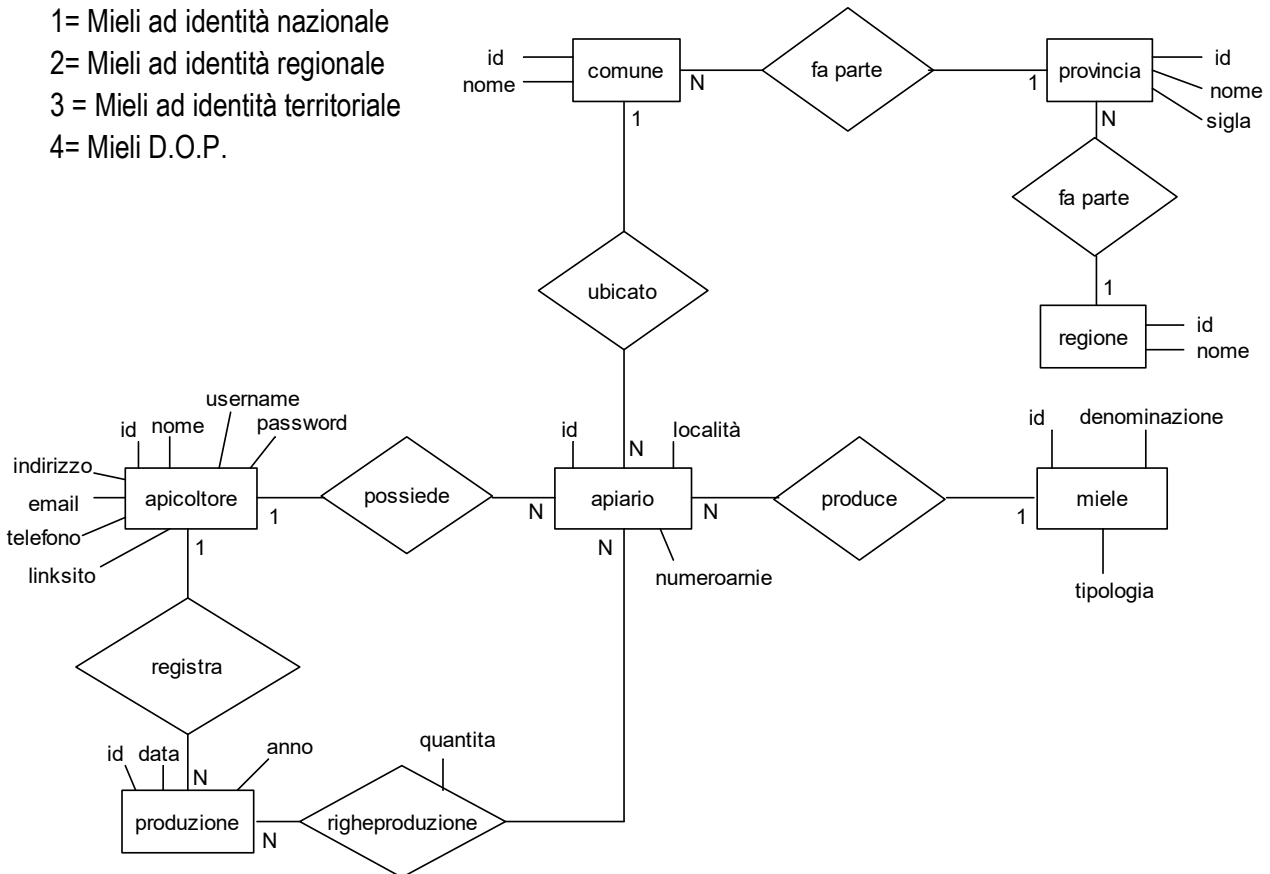
1. ai circa 60000 apicoltori italiani di fornire i dati relativi al miele, distinto per tipologia, prodotto annualmente
2. a chiunque ne abbia interesse, di consultare liberamente la banca dati.

### Schema concettuale (Diagramma E/R)

Per semplicità:

- a) gli attributi delle entità sono quelli più importanti e strettamente necessari per implementare le interrogazioni richieste nella traccia
- b) la tipologia di miele viene considerata come attributo e non come entità, e può assumere i seguenti valori:

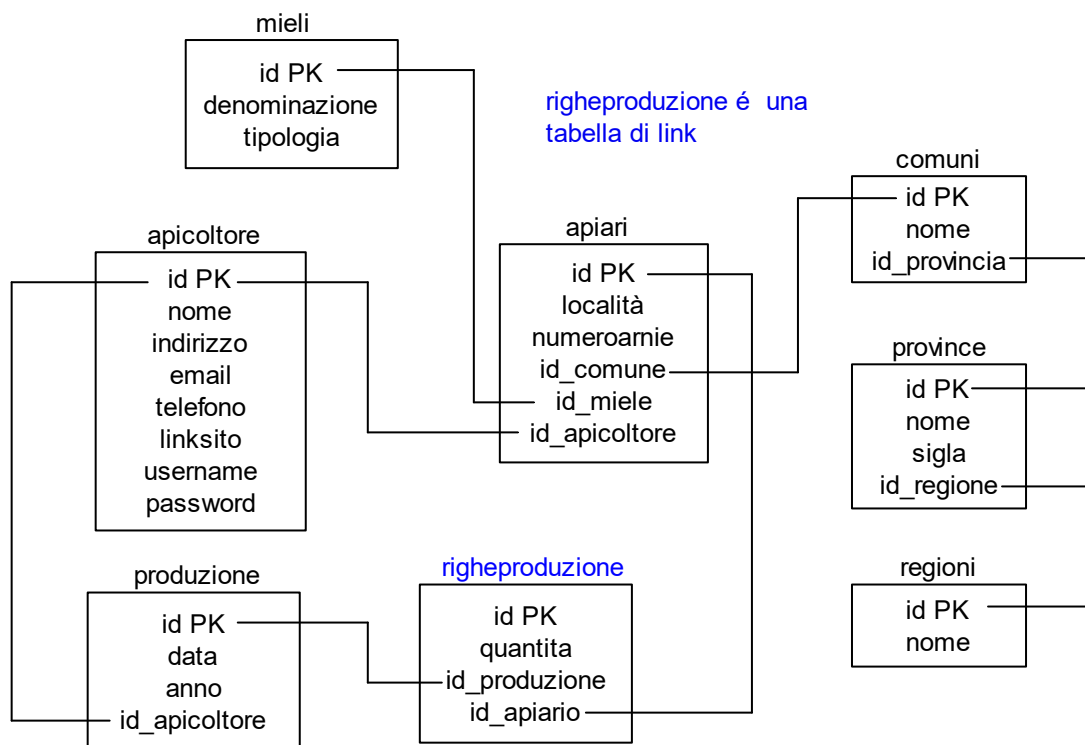
- 1= Mieli ad identità nazionale
- 2= Mieli ad identità regionale
- 3 = Mieli ad identità territoriale
- 4= Mieli D.O.P.



### Lettura del diagramma E/R

Ogni apicoltore possiede N apiari, ogni apiario appartiene ad un apicoltore  
Ogni apiario è ubicato in un comune, in un comune sono ubicati N apiari  
Ogni comune fa parte di una provincia, in ogni provincia ci sono N comuni  
Ogni provincia fa parte di una regione, in ogni regione ci sono N province  
Ogni apiario produce un unico miele, ogni miele è prodotto da N apiari  
Ogni apicoltore registra n produzioni, ogni produzione è registrata da un apicoltore  
Ogni produzione comprende n apiari, ogni apiario riguarda n produzioni

## Schema logico



## Definizione delle relazioni in linguaggio SQL

### Struttura delle tabelle

Si fa riferimento al database MySql Nome del database: mieleitaliano

| tabella    | campi         | key | tipo    | lung | Descrizione/note  |
|------------|---------------|-----|---------|------|---|
| mieli      | id            | PK  | Int     | 4    | Contatore auto_increment  |
|            | denominazione |     | VarChar | 40   | Esempio: Millefiori   |
|            | tipologia     |     | Int     | 1    | Assume i valori da 1 a 4<br>Tipologia 1= Mieli ad identità nazionale<br>Tipologia 2= Mieli ad identità regionale<br>Tipologia 3 = Mieli ad identità territoriale<br>Tipologia 4= Mieli D.O.P. |
| apicoltori | id            | PK  | Int     | 8    | Contatore auto_increment  |
|            | nome          |     | VarChar | 40   |   |
|            | indirizzo     |     | VarChar | 60   |   |
|            | email         |     | VarChar | 30   |   |
|            | telefono      |     | VarChar | 30   |   |
|            | linksito      |     | VarChar | 60   | Link al sito web  |
|            | username      |     | VarChar | 20   |   |
|            | password      |     | VarChar | 32   | Si utilizza la codifica MD5()   |

| tabella         | campi         | key | tipo    | lung | Descrizione/note                |
|-----------------|---------------|-----|---------|------|---------------------------------|
| regioni         | id            | PK  | Int     | 2    | Contatore auto_increment        |
|                 | nome          |     | VarChar | 40   |                                 |
| province        | id            | PK  | Int     | 3    | Contatore auto_increment        |
|                 | nome          |     | VarChar | 40   |                                 |
|                 | sigla         |     | VarChar | 2    |                                 |
|                 | id_regione    | FK  | Int     | 2    | Fa riferimento a regioni.id     |
| comuni          | id            | PK  | Int     | 4    | Contatore auto_increment        |
|                 | nome          |     | VarChar | 40   |                                 |
|                 | id_provincia  | FK  | Int     | 3    | Fa riferimento a province.id    |
| apiari          | id            | PK  | Int     | 8    | Contatore auto_increment        |
|                 | localita      |     | VarChar | 60   |                                 |
|                 | numeroarnie   |     | Int     | 4    |                                 |
|                 | id_comune     | FK  | Int     | 4    | Fa riferimento ad comuni.id     |
|                 | id_miele      | FK  | Int     | 4    | Fa riferimento a mieli.id       |
|                 | id_apicoltore | FK  | Int     | 8    | Fa riferimento ad apicoltori.id |
| produzione      | id            | PK  | Int     | 8    | Contatore auto_increment        |
|                 | data          |     | Date    |      | Data di registrazione           |
|                 | anno          |     | Int     | 4    | Esempio: 2023                   |
|                 | id_apicoltore | FK  | Int     | 8    | Fa riferimento ad apicoltori.id |
| righeproduzione | id            | PK  | Int     | 8    | Contatore auto_increment        |
|                 | quantità      |     | Int     | 6    | Espressa in kg (numero intero)  |
|                 | id_produzione | FK  | Int     | 8    | Fa riferimento a produzioni.id  |
|                 | id_apiaro     | FK  | Int     | 8    | Fa riferimento ad apiari.id     |

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

```
CREATE DATABASE mieleitaliano;
```

```
CREATE TABLE IF NOT EXISTS mieli(
id INT(4) AUTO_INCREMENT PRIMARY KEY,
denominazione VARCHAR(40),
tipologia INT(1));
```

```
CREATE TABLE IF NOT EXISTS apicoltori(
id INT(8) AUTO_INCREMENT PRIMARY KEY,
nome VARCHAR(40),
indirizzo VARCHAR(60),
email VARCHAR(30),
telefono VARCHAR(30),
linksito VARCHAR(60);
username VARCHAR(20),
password VARCHAR(32));
```

```
CREATE TABLE IF NOT EXISTS regioni(  
id INT(2) AUTO_INCREMENT PRIMARY KEY,  
nome VARCHAR(40));
```

```
CREATE TABLE IF NOT EXISTS province(  
id INT(3) AUTO_INCREMENT PRIMARY KEY,  
nome VARCHAR(40),  
sigla VARCHAR(2),  
id_regione INT(2),  
FOREIGN KEY (id_regione) REFERENCES regioni(id));
```

```
CREATE TABLE IF NOT EXISTS comuni(  
id INT(4) AUTO_INCREMENT PRIMARY KEY,  
nome VARCHAR(40),  
id_provincia INT(3),  
FOREIGN KEY (id_provincia) REFERENCES province(id));
```

```
CREATE TABLE IF NOT EXISTS apiari(  
id INT(8) AUTO_INCREMENT PRIMARY KEY,  
localita VARCHAR(60),  
numeroarnie INT(4),  
id_comune INT(4),  
id_miele INT(4),  
id_apicoltore INT(8),  
FOREIGN KEY (id_comune) REFERENCES comuni(id),  
FOREIGN KEY (id_miele) REFERENCES mieli(id),  
FOREIGN KEY (id_apicoltore) REFERENCES apicoltori(id));
```

```
CREATE TABLE IF NOT EXISTS produzioni(  
id INT(8) AUTO_INCREMENT PRIMARY KEY,  
data Date,  
anno INT(4),  
id_apicoltore INT(8),  
FOREIGN KEY (id_apicoltore) REFERENCES apicoltori(id));
```

```
CREATE TABLE IF NOT EXISTS righeproduzione(  
id INT(8) AUTO_INCREMENT PRIMARY KEY,  
quantita INT(4),  
id_produzione INT(8),  
id_apiario INT(8),  
FOREIGN KEY (id_produzione) REFERENCES produzioni(id),  
FOREIGN KEY (id_apiario) REFERENCES apiari(id) );
```

## Interrogazioni espresse in linguaggio SQL

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

### a) l'elenco degli apicoltori che producono miele DOP in una determinata regione

```
Select distinct apicoltori.nome as "Apicoltori Regione Abruzzo che producono Mieli D.O.P."  
From apicoltori, apiari, mieli, regioni, province, comuni  
Where apicoltori.id=apiari.id_apicoltore  
And apiari.id_miele =mieli.id  
And mieli.tipologia=4  
And apiari.id_comune=comuni.id  
And comuni.id_provincia=province.id  
And province.id_regione=regioni.id  
And regioni.nome='Abruzzo';
```

#### Apicoltori Regione Abruzzo che producono Mieli D.O.P.

Guido Lavespa

Remo Labarca

Mario Rossi

### b) il numero complessivo di apiari per ciascuna regione

```
Select regioni.nome as "Regione", count(*) as "Numero complessivo di apiari"  
From apiari, comuni, province, regioni  
Where apiari.id_comune=comuni.id  
And comuni.id_provincia=province.id  
And province.id_regione=regioni.id  
Group by regioni.id;
```

| Regione | Numero complessivo di apiari |
|---------|------------------------------|
| Abruzzo | 7                            |
| Marche  | 1                            |

### c) le quantità di miele prodotto in Italia lo scorso anno per ciascuna delle quattro tipologie

Creiamo una tabella temp con la tipologia e la quantità di miele prodotto nel 2023 e la raggruppiamo per tipologia

```
Create table temp as  
Select righeproduzione.quantita as q, mieli.tipologia as t  
From righeproduzione, produzioni, apiari, mieli  
Where righeproduzione.id_produzione=produzioni.id  
And righeproduzione.id_apiario=apiari.id  
And apiari.id_miele=mieli.id  
And produzioni.anno=2023;
```

| q    | t |
|------|---|
| 550  | 4 |
| 652  | 1 |
| 1234 | 2 |
| 980  | 3 |
| 2500 | 4 |
| 1000 | 1 |
| 543  | 2 |

```
Select t as "Tipologia",sum(q) as "Miele prodotto nel 2023 [Kg]"  
From temp  
Group by t  
Order by t ASC;
```

| Tipologia | Miele prodotto nel 2023 [Kg] |
|-----------|------------------------------|
| 1         | 1652                         |
| 2         | 1777                         |
| 3         | 980                          |
| 4         | 3050                         |

**Progetto di massima della struttura dell'applicazione web per la gestione della realtà sopra presentata, distinguendo le funzionalità da implementare in base ai vari possibili profili di utenza;**

Un'applicazione web:

- d) è un software basato su un'architettura client-server
- e) risiede su un server remoto e non deve essere installata sul dispositivo dell'utente
- f) viene eseguita online su un qualsiasi dispositivo (pc, tablet, smartphone) dotato di un browser e di una connessione di rete (Internet o Intranet aziendale)
- g) è accessibile tramite il protocollo HTTP o HTTPS
- h) offre evidenti vantaggi ma la scrittura del codice presenta difficoltà e criticità dovute alla natura stessa del web che è stato inizialmente concepito per la presentazione di contenuti statici

**Struttura dell'applicazione web**

- **Lato client** (Frontend), ovvero l'interfaccia che consente all'utente di interagire con l'applicazione. Per la sua realizzazione utilizziamo principalmente HTML5, CSS e Javascript. Per rendere le pagine interattive e professionali utilizziamo AJAX
- **Lato server** (Backend), ovvero il cuore dell'applicazione che si connette al database per leggere e scrivere i dati, elabora le richieste del client e restituisce le risposte. Tra i molteplici linguaggi di programmazione disponibili, scegliamo di utilizzare PHP
- **Database** che memorizza e gestisce i dati dell'applicazione web. Utilizziamo il Database MySQL

Riguardo al server remoto che ospita l'applicazione, considerando il numero degli apicoltori coinvolti (circa 60000) e dei potenziali utenti pubblici, è richiesta un'infrastruttura molto potente e performante che sia in grado di garantire la disponibilità, affidabilità e sicurezza dei dati necessarie.

**Principali funzionalità previste**

Oltre ai due profili indicati nella traccia, relativi agli apicoltori opportunamente registrati e agli utenti pubblici che accedono liberamente ad alcune pagine di consultazione dei dati, occorre tener conto del profilo dei tecnici che gestiscono i dati delle tabelle comuni (mieli, apicoltori, comuni, province, regioni) e degli amministratori dell'applicazione web

- gli utenti pubblici, liberamente:
  - consultano le pagine di presentazione della banca dati, delle caratteristiche delle diverse varietà di miele, delle zone di produzione
  - tramite la mappa cliccabile dell'Italia visualizzano le informazioni principali dei produttori di miele di una regione o di una provincia ed eventualmente accedono ai loro siti web (per ottenere maggiori informazioni, fare acquisti online ecc.)
  - ...
- gli apicoltori :
  - accedono tramite una procedura di login (livello di accesso 1)
  - creano l'account utente
  - gestiscono i dati riguardanti il proprio profilo e gli apiari di cui dispongono
  - inseriscono i dati relativi alle loro produzioni annuali
  - visualizzano report e statistiche
  - ...

- i tecnici:
  - accedono tramite una procedura di login (livello di accesso 2)
  - gestiscono i dati delle tabelle comuni (inserimento, modifica, cancellazione)
  - forniscono assistenza clienti e risolvono i problemi segnalati dagli apicoltori riguardo agli archivi
  - ...
- gli amministratori:
  - accedono tramite una procedura di login (livello di accesso 3)
  - assicurano che l'applicazione funzioni correttamente
  - gestiscono i livelli di accesso degli apicoltori
  - forniscono assistenza clienti e risolvono i problemi segnalati dagli apicoltori riguardo all'applicazione
  - ...

In base al livello di accesso, l'applicazione consente o nega ai profili di utenza di accedere alle varie pagine.

L'applicazione prevede altre funzionalità quali ricerche avanzate, filtri, ordinamenti, report, statistiche e grafici che consentono di organizzare al meglio i dati relativi alle produzioni di miele. Inoltre sono previste le opzioni di utilità del recupero password e della chat in tempo reale tra apicoltori e assistenza clienti.

### **Una parte significativa dell'applicazione web che consente l'interazione con la base di dati, utilizzando appropriati linguaggi a scelta sia lato client che lato server**

Scegliamo di codificare l'opzione di "Inserimento miele" (a questa funzionalità possono accedere gli utenti registrati con livello di accesso 2)

Il codice è composto dal file **insmiele.html** (lato client) e dal file **salvamiele.php** (lato server)

Per semplicità il codice proposto è molto essenziale e non viene utilizzato alcun css; manca inoltre la parte che verifica se l'utente è autorizzato o meno ad accedere alla pagina stessa.

Database MySql: **mieleitaliano**

Tabella: **mieli**

| Nome  | Tipo        |
|---|-------------|
| <b>id</b>  | int         |
| <b>denominazione</b>  | varchar(40) |
| <b>tipologia</b>  | int         |

## Lato client: HTML5, JavaScript, AJAX

### insmiele.html

```
<!DOCTYPE html>
<html lang="it">
<head>
  <title>Inserimento mieli</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <script src="http://code.jquery.com/jquery-latest.js"></script>
</head>
<script type="text/javascript">
$(document).ready(function() {
  $('#reset').click(function(){
    $('#risposta').empty(); //il reset deve pulire anche div risposta
  });
  $('#submit').click(function(){
    var den=$("#den").val();
    var tip=$("#tip").val();
    if(den==""){
      alert("Dati incompleti: inserire la denominazione");
      return false;
    }
    if(tip==0){
      alert("Dati incompleti: selezionare la tipologia");
      return false;
    }
    $.ajax({ //chiamata AJAX di jquery
      type: "POST",
      url: "salvamiele.php",
      data: "den="+den+"&tip="+tip, dataType: "html",
      success: function (msg){
        $('#risposta').html(msg);
      }
    });
    return false;
  });
});
</script>
<body style="background-color:#fafafa">
  <h1>Immissione miele</h1>
  <form name="form1" id=" form" method="post" action="">
    <label>Denominazione</label><br/>
    <input type="text" id="den" style="width:200px" ><br/><br/>
    <label>Tipologia</label><br/>
    <select name="tipologie" id="tip" style="width:200px">
      <option value="0">Scegli</option>
      <option value="1">Mieli ad identità nazionale</option>
      <option value="2">Mieli ad identità regionale</option>
      <option value="3">Mieli ad identità territoriale</option>
      <option value="4">Mieli D.O.P.</option>
    </select><br/><br/>
    <button type="submit" id="submit" style="width:62px">Salva</button>
    <button type="reset" id="reset" style="width:62px">Reset</button>
    <button type="button" id="home" style="width:62px">Home</button><br/><br/>
    <div id="risposta"> </div><br/>
  </form>
</body>
</html>
```

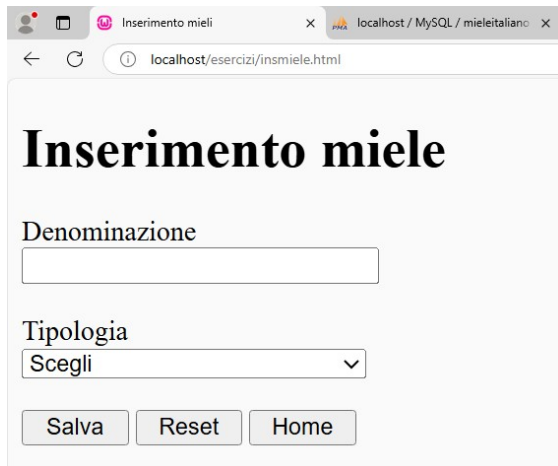
## Lato server: PHP

### salvamiele.php

```
<?php
    $conn=mysqli_connect("localhost","root","mypassword","mieleitaliano");
    $denominazione=mysqli_real_escape_string($conn,$_POST["den"]);
    // buona pratica vs MySql iniecton
    $tipologia=$_POST["tip"];
    $messaggio="";
    // verifico se la denominazione esiste già nella tabella mieli
    $s="Select * from mieli where denominazione='$denominazione'";
    $q=mysqli_query($conn,$s);
    if (mysqli_num_rows($q)==0){ // se il miele non esiste
        $s="Insert into mieli (denominazione,tipologia)
        Values('$denominazione','$tipologia)";
        $q=mysqli_query($conn,$s);
        $messaggio="Il miele '". $denominazione. "' è stato salvato
correttamente";
    }
    else{
        $messaggio="Il miele '". $denominazione. "' esiste già";
    }
    if($q){ // se la query è stata eseguita
        echo $messaggio;
    }
    else {
        echo "Inserimento non riuscito, ripetere l'operazione";
    }
    mysqli_close($conn);
?>
```

## Test e spiegazione 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.



The screenshot shows a web browser window with the URL localhost/esercizi/insmiele.html. The page title is "Inserimento miele". There is a text input field for "Denominazione" and a dropdown menu for "Tipologia" with "Scegli" selected. Below the form are three buttons: "Salva", "Reset", and "Home".



The screenshot shows the same web browser window. The "Denominazione" field now contains "Miele delle colline teramane". The "Tipologia" dropdown menu is open, showing a list of options: "Scegli", "Mieli ad identità nazionale", "Mieli ad identità regionale", "Mieli ad identità territoriale", and "Mieli D.O.P.". The "Scegli" option is currently selected.

- Lato client: l'utente (registrato con livello di accesso 2) all'interno del form di inserimento digita la denominazione del miele e, tramite una select, seleziona la tipologia. Preme il bottone Salva. JavaScript dopo aver controllato la completezza dei dati, tramite una richiesta asincrona AJAX, invia i valori al server PHP.
- Lato server: salvamiele.php riceve i valori di "denominazione" e "tipologia" tramite le variabili \$\_POST["..."] e dopo aver controllato che il miele non sia stato già precedentemente inserito, esegue la query di insert nella tabella mieli e restituisce una risposta
- Lato client: la risposta viene mostrata nell'elemento `<div id="risposta"> </div>` del form senza che la pagina HTML5 venga ricaricata



The screenshot shows the web browser window after the form submission. The "Denominazione" field contains "Miele delle colline teramane" and the "Tipologia" dropdown menu is set to "Mieli D.O.P.". Below the form, a message is displayed: "Il miele 'Miele delle colline teramane' è stato salvato correttamente".

## Soluzione seconda parte

- I. In relazione al tema proposto nella prima parte si realizzi, in un linguaggio a scelta, una porzione di codice significativa della pagina dinamica che consente all'utente di ottenere l'elenco degli apiari di una certa regione, selezionando sulla pagina web la regione di interesse a partire da una rappresentazione in forma grafica del territorio italiano.

Quantunque la traccia richieda di realizzare solo una porzione di codice, propongo agli studenti la soluzione completa della pagina dinamica descritta nel quesito affrontando in particolare il tema delle mappe cliccabili. Selezionare una regione a partire da una rappresentazione in forma grafica del territorio italiano è concettualmente facile perché HTML mette a disposizione i tag e i relativi attributi che consentono di creare mappe di immagini cliccabili, ma è sicuramente laborioso e complesso. Per semplificare il lavoro riformuliamo il quesito: "... si realizzi in un linguaggio a scelta, il codice della pagina dinamica che consente all'utente di ottenere l'elenco degli apiari di una determinata provincia abruzzese selezionandola a partire da una rappresentazione grafica della regione Abruzzo".

I tag HTML che consentono di creare una mappa di immagine cliccabile sono `<img>` `<map>` e `<area>`.

Nel nostro caso specifico :

`<img>` consente di inserire l'immagine della regione Abruzzo nella pagina HTML

`<map>` genera la mappa HTML relativa all'immagine della regione Abruzzo

`<area>` genera le aree delle 4 province sensibili al click del mouse



Partendo dunque dall'immagine abruzzo.jpg che abbiamo a disposizione, creiamo una mappa cliccabile delle 4 province della regione Abruzzo

```

  <map name="Map">
    <area shape="rect" coords="140,345,270,405" href="apiari.php?prov='AQ'"/>
    <area shape="rect" coords="200,100,325,170" href="apiari.php?prov='TE'"/>
    <area shape="rect" coords="300,200,413,260" href="apiari.php?prov='PE'"/>
    <area shape="rect" coords="450,360,555,420" href="apiari.php?prov='CH'"/>
  </map>
```

Il tag fondamentale delle mappe cliccabili, è `<area>` con gli attributi `href` e `shape`.

- `href` è il riferimento ipertestuale alla pagina php che stampa l'elenco degli apiari della provincia selezionata
- `shape` definisce la forma dell'area cliccabile che può essere di tre tipi:
  - `rect` – area rettangolare
    - l'attributo `coords` specifica 2 coppie di valori numerici, ovvero le coordinate `x1,y1` del punto corrispondente all'angolo superiore sinistro e le coordinate `x2,y2` del punto corrispondente all'angolo inferiore destro.
  - `circle`: area circolare
    - l'attributo `coords` specifica tre valori: le coordinate `x,y` del centro del cerchio e la lunghezza del raggio in pixel;

- poly: area poligonale, che dovremmo usare per "seguire" i confini delle province
  - l'attributo coords specifica le coordinate x,y di tutti i punti che definiscono il contorno dell'area. Più numerosi sono i punti della poligonale, migliore è l'approssimazione dell'area cliccabile alla mappa

Nel nostro caso la mappa è molto semplice, lo shape è di tipo "rect" e le zone sensibili al passaggio del mouse riguardano le aree rettangolari attorno ai nomi dei capoluoghi .

Per individuare le coordinate x,y dello shape, possiamo utilizzare un comune programma di grafica.

Facendo riferimento a paint o a paint.net, apriamo l'immagine abruzzo.jpg.

Sulla barra inferiore il programma ci restituisce:

- la larghezza e l'altezza in pixel dell'immagine
- le coordinate x, y del punto in cui è posizionato il mouse. Muovendo il mouse all'interno dell'immagine siamo in grado di individuare le coordinate di tutti i punti che desideriamo

Il codice della pagina è composto dal file **apiari.html** (lato client) e dal file **apiariprov.php** (lato server)

## Lato client: HTML5, JavaScript, AJAX

### apiari.html

Per la creazione della mappa della regione Abruzzo suddivisa nelle 4 province, utilizziamo le coordinate viste nella pagina precedente, ma modifichiamo il codice in maniera tale da poter utilizzare JS e AJAX

```
<!DOCTYPE html>
<html lang="it">
<head>
  <title>Mappa Abruzzo</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <script src="http://code.jquery.com/jquery-latest.js"></script>
</head>
<script type="text/javascript">
$(document).ready(function() {
  $('#reset').click(function(){
    $('#risposta').empty();
  });
  $(".AQ").on("click", function(e){
    e.preventDefault();
    //la riga sopra consente di by-passare l'azione HREF di
    //default nelle mappe cliccabili
    var prov="AQ";
    $.ajax({
      // la chiamata AJAX consente di stampare la lista degli apiari della
      // provincia selezionata senza ricaricare la pagina
      type: "POST",
      url: "apiariprov.php",
      data: "prov="+prov,
      success: function (msg){
        $('#risposta').html(msg);
      }
    });
  });
  return false;
});
```

```
$(".TE").on("click", function(e){
    e.preventDefault();
    var prov="TE";
    $.ajax({
        type: "POST",
        url: "apiariprov.php",
        data: "prov="+prov,
        success: function (msg){
            $('#risposta').html(msg);
        }
    });
});
return false;
});
$(".PE").on("click", function(e){
    e.preventDefault();
    var prov="PE";
    $.ajax({
        type: "POST",
        url: "apiariprov.php",
        data: "prov="+prov,
        success: function (msg){
            $('#risposta').html(msg);
        }
    });
});
return false;
});
$(".CH").on("click", function(e){
    e.preventDefault();
    var prov="P";
    $.ajax({
        type: "POST",
        url: "apiariprov.php",
        data: "prov="+prov,
        success: function (msg){
            $('#risposta').html(msg);
        }
    });
});
return false;
});
});
</script>
<body>
<h1>Apiari in Abruzzo</h1>
<form name="form1" id=" form" method="post" action="">
<h2>Scegli la provincia:</h2>
    
    <map name="Map">
        <area class="AQ" shape="rect" coords="140,345,270,405" href="" />
        <area class="TE" shape="rect" coords="200,100,325,170" href="" />
        <area class="PE" shape="rect" coords="300,200,413,260" href="" />
        <area class="CH" shape="rect" coords="450,360,555,420" href="" />
    </map>
    <br/>
    <button type="reset" id="reset" style="width:62px">Reset</button>
    <button type="button" id="home" style="width:62px">Home</button><br/><br/>
    <div id="risposta"> </div><br/>
</form>
</body>
</html>
```

## Lato server: PHP

### apiariprov.php

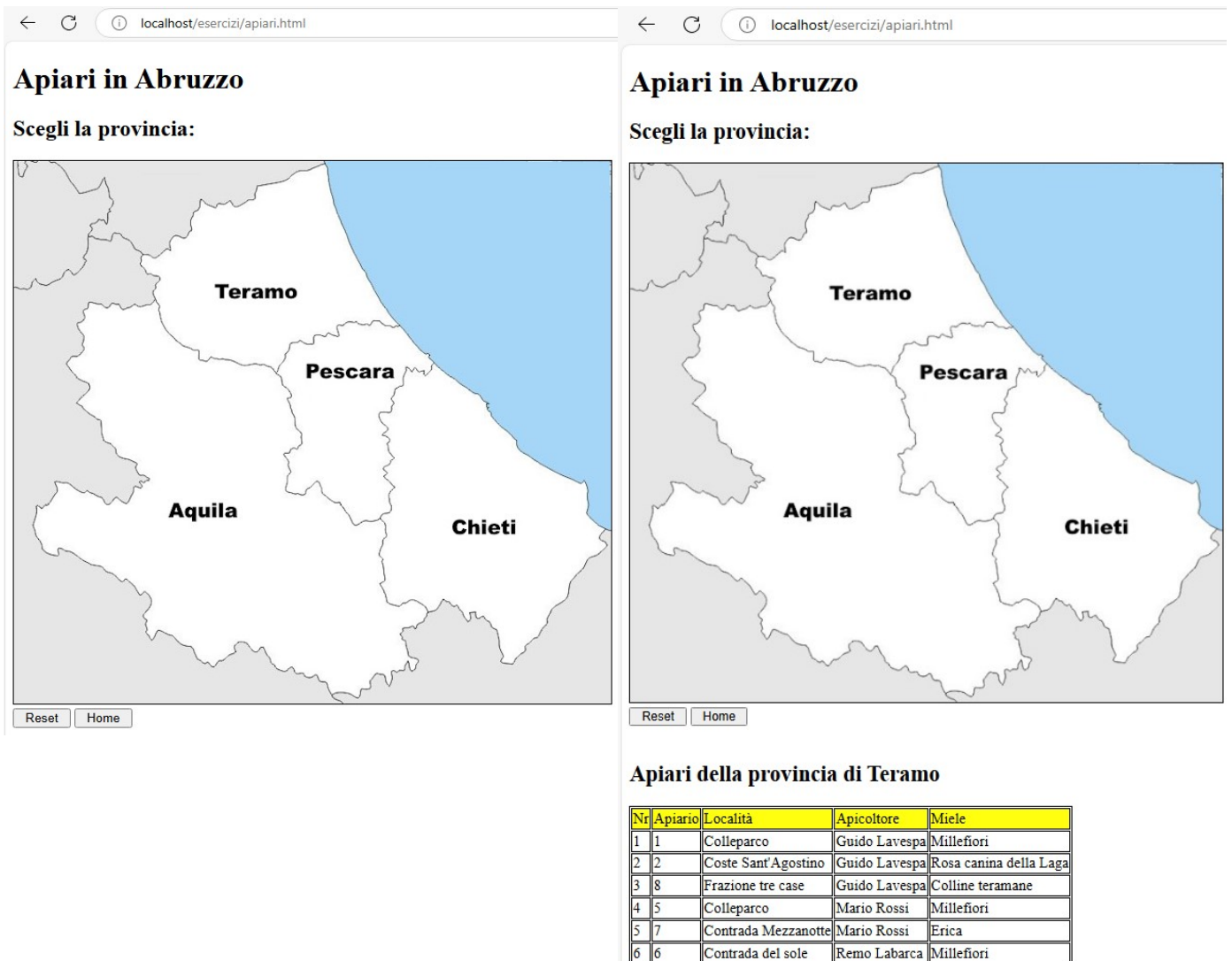
```
<?php
$conn=mysqli_connect("localhost","root","mypassword","mieleitaliano");
$sigla=mysqli_real_escape_string($conn,$_POST["prov"]);
// buona pratica vs MySql iniecton
$s="Select nome from province where sigla='$sigla'";
$q=mysqli_query($conn,$s);
$r = mysqli_fetch_array($q);
$provincia=$r[0];
$ris="<h2>Apiari della provincia di '$provincia'</h2>";
$s="Select apiari.id,apiari.localita,apicoltori.nome,mieli.denominazione
from apiari,comuni, province,apicoltori,mieli
where province.sigla='$sigla'
and apiari.id_comune=comuni.id
and comuni.id_provincia=province.id
and apiari.id_apicoltore=apicoltori.id
and apiari.id_miele=mieli.id
order by apicoltori.nome";
$q=mysqli_query($conn,$s);

if (mysqli_num_rows($q)==0){

    $ris.="Non ci sono apiari";
}
else{
    $ris.="<style>table, th, td {border: 1px solid black;}</style>";
    $k=1;
    $ris.="<table>
        <tr bgcolor='#FFFF0F'>
            <td>Nr</td><td>Apiario</td><td>Località</td>
            <td>Apicoltore</td><td>Miele</td>
        </tr>";
    while($r = mysqli_fetch_array($q)) {
        $k++;
        $ris.="<tr><td>$k</td>";
        $ris.="<td>$r[0]</td> <td>$r[1]</td>
            <td>$r[2]</td> <td>$r[3]</td></tr>";
    }
    $ris.="</table><br/>";
}
echo $ris;
mysqli_close($conn);
?>
```

## Test e spiegazione 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.



The image displays two browser windows side-by-side, both showing the URL `localhost/esercizi/apiari.html`. The left window shows a map of the Abruzzo region with the provinces of Teramo, Pescara, Aquila, and Chieti labeled. Below the map are two buttons: "Reset" and "Home". The right window shows the same map, but with an additional table titled "Apiari della provincia di Teramo" below it. The table contains the following data:

| Nr | Apiario | Località            | Apicoltore    | Miele                  |
|----|---------|---------------------|---------------|------------------------|
| 1  | 1       | Colleparco          | Guido Lavespa | Millefiori             |
| 2  | 2       | Coste Sant'Agostino | Guido Lavespa | Rosa canina della Laga |
| 3  | 8       | Frazione tre case   | Guido Lavespa | Colline teramane       |
| 4  | 5       | Colleparco          | Mario Rossi   | Millefiori             |
| 5  | 7       | Contrada Mezzanotte | Mario Rossi   | Erica                  |
| 6  | 6       | Contrada del sole   | Remo Labarca  | Millefiori             |

- Lato client: l'utente clicca sulla provincia e JavaScript, tramite una richiesta asincrona AJAX, invia la sigla della provincia al server PHP
- Lato server: `apiariprov.php` riceve il valore della sigla tramite la variabile `$_POST["prov"]`, esegue la query che consente di ottenere l'elenco degli apiari della regione selezionata e stampa una tabella come risposta
- Lato client: la risposta viene mostrata nell'elemento `<div id="risposta"> </div>` del form senza che la pagina HTML5 venga ricaricata

- II. In relazione al tema proposto nella prima parte, prevedendo eventuali integrazioni della base di dati, si realizzi in un linguaggio a scelta la pagina dinamica che visualizza l'elenco prodotto dalla query a) del punto 4 della prima parte, riportando in aggiunta per ciascun apicoltore l'eventuale link al suo sito web e la mail.

La query a cui si riferisce il quesito deve restituire a) l'elenco degli apicoltori che producono miele DOP in una determinata regione.

La pagina dinamica che vogliamo realizzare è composta dal file **ele\_apicoltori.php**, tramite il quale selezioniamo la regione e la tipologia di miele (l'estensione è php perché occorre utilizzare il linguaggio PHP per popolare la select regioni), e dal file **sel\_apicoltori.php** che esegue la query e restituisce l'elenco desiderato.

Coerentemente con i codici proposti nelle pagine precedenti, anche in questo caso utilizziamo AJAX che consente lo scambio in background dei dati tra i due file ed evita il ricaricamento della pagina HTML. Con AJAX, JavaScript inoltra una richiesta GET o POST al server, riceve ed elabora la risposta e aggiorna la pagina corrente. L'utente non si accorge che qualcosa è stato trasmesso al server e l'applicazione risulta più fluida, veloce e interattiva.

### ele\_apicoltori.php

```
<!DOCTYPE html>
<html lang="it">
<head>
  <title>Apicoltori per regione</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <script src="http://code.jquery.com/jquery-latest.js"></script>
</head>
<script type="text/javascript">
  $(document).ready(function() {

    $('#reset').click(function(){
      $("#risposta").empty();

    });

    $('#submit').click(function(){
      var reg=$("#reg").val();
      var tip=$("#tip").val();
      if(reg==0)
      {
        alert("Dati incompleti: selezionare la regione");
        return false;
      }
      if(tip==0)
      {
        alert("Dati incompleti: scegliere la tipologia");
        return false;
      }
    }
  }
);
```

```
//chiamata ajax di jquery.
$.ajax({
  type: "POST",
  url: "sel_apicoltori.php", data: "reg="+reg+"&tip="+tip,
  dataType: "html",
  success: function (msg){
    $('#risposta').html(msg);
  }
});
return false;
});
});
</script>

<body style="background-color:#fafafa">
  <h1>Elenco apicoltori per regione</h1>
  <form name="form1" id=" form" method="post" action="">
    <label>Regione</label><br/>
    <select name="regioni" id="reg" style="width:200px">
      <?php
        $conn=mysqli_connect("localhost","root","", "mieleitaliano");
        //si connette al database
        $s = "Select * From regioni Order By nome";
        // Legge le righe della tabella regioni
        $query=mysqli_query($conn,$s);
        // popola la select regioni con i dati della tabella
        echo '<option value="0">Seleziona la regione</option>';
        while($r = mysqli_fetch_array($query)) {
          echo '<option value="'. $r[0].'">'. $r[1].'</option>';
        }
        mysqli_close($conn);
      ?>
    </select><br/><br/>
    <label>Tipologia</label><br/>
    <select name="tipologie" id="tip" style="width:200px">
      <option value="0">Scegli la tipologia</option>
      <option value="1">Mieli ad identità nazionale</option>
      <option value="2">Mieli ad identità regionale</option>
      <option value="3">Mieli ad identità territoriale</option>
      <option value="4">Mieli D.O.P.</option>
    </select><br/><br/>
    <button type="submit" id="submit" style="width:62px">Submit</button>
    <button type="reset" id="reset" style="width:62px">Reset</button>
    <button type="button" id="home" style="width:62px">
      Home</button><br/><br/>
    <div id="risposta"> </div><br/>
  </form>
</body>
</html>
```

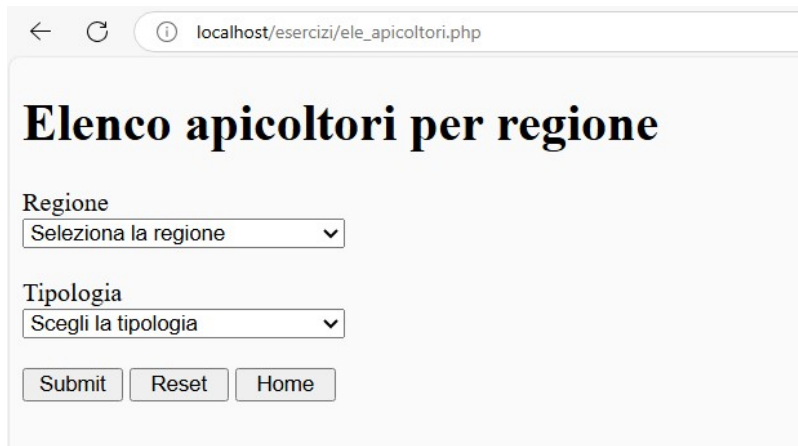
## sel\_apicoltori.php

```
<?php
$conn=mysqli_connect("localhost","root","mypassword","mieleitaliano");
$reg=$_POST["reg"];
$tip=$_POST["tip"];
// recupero il nome della regione
$s="Select nome from regioni where id=$reg";
$q=mysqli_query($conn,$s);
$r= mysqli_fetch_array($q);
$regione=$r[0];
// Non avendo utilizzato una tabella...
if($tip==1)$tipologia="Mieli ad identità nazionale";
if($tip==2)$tipologia="Mieli ad identità regionale";
if($tip==3)$tipologia="Mieli ad identità territoriale";
if($tip==4)$tipologia="Mieli D.O.P.";
// -----
$ris="<h3>Apicoltori della regione $regione del miele '$tipologia'</h3>";
$s="Select distinct apicoltori.nome,apicoltori.email,apicoltori.linksito
    From apicoltori, apiari, mieli,regioni, province,comuni
    Where apicoltori.id=apiari.id_apicoltore
    And apiari.id_miele =mieli.id
    And mieli.tipologia=$tip
    And apiari.id_comune=comuni.id
    And comuni.id_provincia=province.id
    And province.id_regione=regioni.id
    And regioni.id=$reg";
$q=mysqli_query($conn,$s);
$nr=mysqli_num_rows($q);
if ($nr==0){

    $ris.="Nessun apicoltore trovato";
}
else{
    $ris.="<style>table, th, td {border: 1px solid black;}</style>";
    $k=0;
    $ris.="<table>
        <tr bgcolor='#FFFF0F'>
            <td>Nr</td><td>Apicoltore</td><td>Email</td>
            <td>Sito web</td>
        </tr>";
    while($r = mysqli_fetch_array($q)) {
        $k++;
        $ris.="<tr><td>$k</td>";
        $ris.="<td>$r[0]</td> <td>$r[1]</td>
            <td>$r[2]</td></tr>";
    }
    $ris.="</table><br/>";
}
echo $ris;
mysqli_close($conn);
?>
```

## Test e spiegazione 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.



| Nr | Apicoltore    | Email                 | Sito web                  |
|----|---------------|-----------------------|---------------------------|
| 1  | Guido Lavespa | guidolavespa@hhh.it   | www.guidolavespa-mieli.it |
| 2  | Mario Rossi   | infomielrossi@xxx.com | www.mielrossimario.eu     |
| 3  | Remo Labarca  | remolabarca@xxx.it    | www.mieliremolabarca.it   |

- Lato client: l'utente tramite le select "regioni" e "tipologie" seleziona la regione e la tipologia del miele prodotto. La select "regioni" viene popolata tramite codice PHP prendendo i dati dalla tabella "regioni" del database MySQL "mieleitaliano". JavaScript, tramite una richiesta asincrona AJAX, invia l'id della regione e l'id della tipologia al server PHP.
- Lato server: sel\_apicoltori.php riceve i valori dell'id regione e dell'id tipologia attraverso le variabili \$\_POST["reg"] e \$\_POST["tip"], esegue la query che consente di ottenere l'elenco desiderato e stampa una tabella come risposta
- Lato client: la risposta viene mostrata nell'elemento `<div id="risposta"> </div>` del form senza che la pagina HTML5 venga ricaricata

III. In considerazione del fatto che il protocollo http è "stateless", il candidato descriva le possibili modalità di propagazione dei dati fra successive pagine web consultate in un sito, al fine di assicurarne la persistenza.

Il protocollo HTTP è "stateless", ovvero senza stato; significa che ogni richiesta http dal client al server è indipendente dalle altre e non mantiene informazioni sulle richieste precedenti. Una comunicazione stateless consente di ottimizzare le prestazioni del web ma rappresenta un problema in tutti i casi in cui, per i più disparati motivi, occorre propagare i dati fra le successive pagine web.

Facendo riferimento ad esempio ad un sito di e-commerce, l'utente si deve inizialmente autenticare e da quel momento in poi il suo stato deve essere mantenuto nelle successive pagine web. Allo stesso modo i prodotti che l'utente inserisce nel carrello degli acquisti devono essere mantenuti nelle pagine successive in cui l'acquisto viene perfezionato e poi concluso.

Per propagare i dati fra successive pagine web in un sito, possiamo utilizzare le seguenti tecniche:

1. Invio di dati tramite un form con i metodi GET e POST
2. Utilizzo di campi nascosti in un form
3. Utilizzo di query string
4. Memorizzazione dei dati in una sessione

Invio di dati tramite un form con i metodi POST e GET

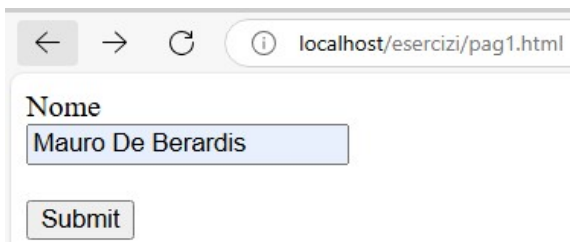
Quando si usa un form HTML con il metodo POST o il metodo GET, i dati inseriti dall'utente vengono inviati al server come parte della richiesta HTTP.

pag1.html Invia i dati in un form

```
<form action="pag2.php" method="POST">
  <label>Nome</label><br/>
  <input type="text" id="nome" name="nome">
  <br/><br/>
  <input type="submit" value="Submit">
</form>
```

pag2.php Recupera i dati

```
<?php
  $nome = $_POST['nome'];
  echo "Utente: ".$nome;
?>
```



localhost/esercizi/pag1.html

Nome



localhost/esercizi/pag2.php

Utente: Mauro De Berardis

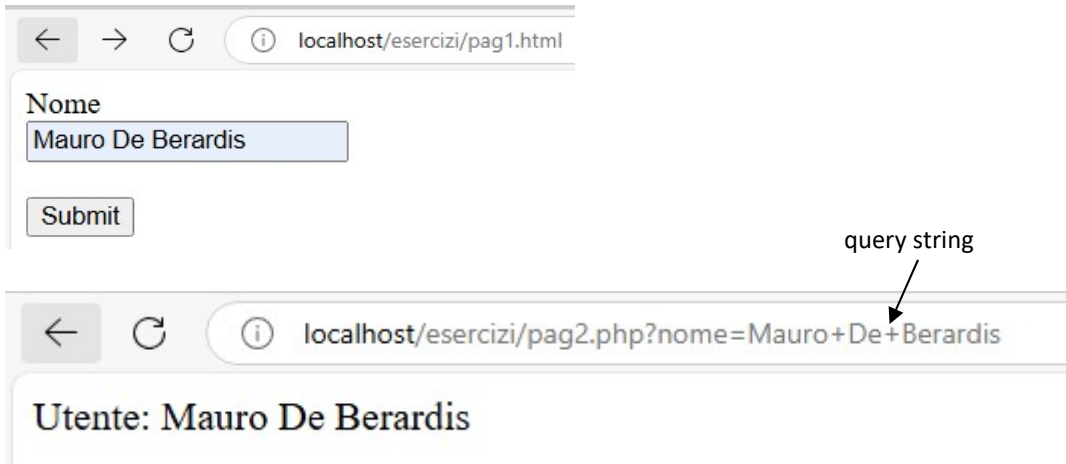
Diversamente dal metodo POST in cui i dati non sono visibili nell'URL, se si utilizza il metodo GET i dati sono inclusi nell'URL sotto forma di query string

pag1.html Invia i dati di un form

```
<form action="pag2.php" method="GET">
  <label>Nome</label><br/>
  <input type="text" id="nome" name="nome">
  <br/><br/>
  <input type="submit" value="Submit">
</form>
```

pag2.php Recupera i dati

```
<?php
  $nome = $_GET['nome'];
  echo "Utente: ".$nome;
?>
```



I dati di un form inviati con POST o GET non vengono conservati nelle richieste successive. La propagazione automatica dei dati è possibile, come vedremo, utilizzando le sessioni

### Utilizzo di campi nascosti in un form

Spesso per propagare i dati tra le pagine si utilizzano i campi nascosti. Ad esempio pag2.php per propagare il nome ricevuto da pag1.html alla pagina successiva pag3.php, utilizza il campo di tipo `hidden` ovvero nascosto

```
<?php
  $nome = $_POST['nome'];
  echo "Utente: ".$nome;
  <form action="pag3.php" method="POST">
    ...
    ...
    <input type="hidden" name="nome" value="<?php $nome; ?>">
    <input type="submit" value="submit">
  </form>
?>
```

### Utilizzo di query string

L'uso di query string, già vista nell'invio dati di un form con il metodo GET, consente di propagare i dati tra pagine passando i dati tramite una stringa che segue il simbolo ? dopo l'url.

Esempio:

pag1.html      Invia i dati tramite query string

```
<a href="pag2.php?nome=Mauro De Berardis &nazione=Italia">Vai a Pag2</a>
```

pag2.php      recupera i dati con GET

```
<?php
    $nome = $_GET['nome'];
    $nazione=$_GET['nazione'];
    echo "Utente: ".$nome;
    echo "Nazione: ".$nazione;
?>
```



Ovviamente i dati possono essere inviati anche ad un'altra pagina HTML e recuperati con le funzioni JavaScript

### Memorizzazione dei dati in una sessione

Per spiegare come funziona una sessione e come si utilizza, facciamo riferimento al meccanismo che consente di propagare la username e il livello di accesso di un utente che si è autenticato attraverso una procedura di login, nelle varie pagine dell'applicazione web che gestisce il miele italiano.

Dopo che l'utente è stato autenticato nella pagina iniziale di login, vengono create due variabili di sessione

```
<?php
    session_start();// Inizializza la sessione;
    // session_start() è necessaria ogni volta che si vuole utilizzare una sessione
    $autenticato=false;
    // codice che autentica le credenziali fornite dall'utente e gli assegna un livello di accesso
    if($autenticato){
        $_SESSION['username'] = "maurodeb";
        $_SESSION['livello']=2;
    }
?>
```

In tutte le altre pagine dell'applicazione web, le variabili di sessione vengono recuperate e gestite

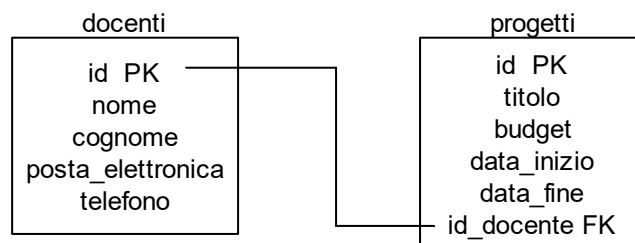
```
<?php
session_start();
if (!isset($_SESSION['username'])) {
    // se non esiste $_SESSION['username'] vuol dire che l'utente non si è autenticato
    // e viene rediretto alla pagina di login
    header('Location:login.php');
    exit;
}
$username=$_SESSION['username'];
$livello=$_SESSION['livello'];
// segue codice della pagina che tiene conto dell'utente e del suo livello di accesso
?>
```

- IV. Si supponga che siano già state create le tabelle corrispondenti al seguente schema logico  
Progetti (id, titolo, budget, data\_inizio, data\_fine)  
Docenti (id, nome, cognome, posta\_elettronica, telefono)

Si utilizzino opportuni comandi SQL per:

- aggiungere nella tabella Progetti il campo che identifica il docente responsabile di ciascun progetto descritto nella tabella corrispondente
- rinominare nella tabella Docenti il campo "posta\_elettronica" in "e-mail"

- a) lo schema logico che lega le tabelle progetti e docenti è il seguente:



In questo schema ogni docente realizza n progetti, ogni progetto è realizzato da un docente. La relazione tra docenti e progetti è 1-a-N e il campo della tabella progetti che identifica il docente è la chiave esterna (FK) id\_docente che fa riferimento alla chiave primaria (PK) id della tabella docenti.

Per aggiungere nella tabella progetti il campo che identifica il docente occorre:

- creare una nuova colonna id\_docente  
**ALTER TABLE progetti ADD COLUMN id\_docente INT NOT NULL;**
- assegnare una chiave esterna alla colonna appena creata  
**ALTER TABLE progetti ADD FOREIGN KEY(id\_docente) REFERENCES docenti(id)**

- b) ALTER TABLE docenti RENAME COLUMN posta\_elettronica TO email  
Inserendo e-mail al posto di email, come richiesto nel quesito, ed eseguendo la query su phpMyAdmin 5.2.0 di Wampserver 3.3.0 64 bit si verifica un errore che rinuncio ad indagare considerando che l'uso del trattino "-" è sconsigliato nell'assegnazione dei nomi dei campi delle tabelle MySQL