Esame di stato 2002 Banca del tempo

Traccia

Un associazione Banca del Tempo vuole realizzare una base di dati per registrare e gestire le attività dell'associazione.
La Banca del Tempo (BdT) indica uno di quei sistemi organizzati di persone che si associano per scambiare servizi e/o saperi, attuando un aiuto reciproco.
Attraverso la BdT le persone mettono a disposizione il proprio tempo per determinate prestazioni (effettuare una piccola riparazione in casa, preparare una torta, conversare in lingua straniera, ecc) aspettando di ricevere prestazioni da altri Non circola denaro, tutte le prestazioni sono valutate in tempo, anche le attività di segreteria Le prestazioni sono suddivise in categorie (lavori manuali,tecnologie, servizi di trasporto, bambini, attività sportive, ecc).
Chi dà un ora del suo tempo a qualunque socio, riceve un ora di tempo da chiunque faccia parte della BdT La base di dati dovrà mantenere le informazioni relative ad ogni prestazione (quale prestazione, da chi è stata erogata, quale socio ha ricevuto quella prestazione, per quante ore e in quale data) per consentire anche interrogazioni di tipo statistico.
Il territorio di riferimento della BdT è limitato (un quartiere in una grande città o un piccolo comune) ed è suddiviso in zone; la base di dati dovrà contenere la mappa del territorio e delle singole zone, in forma grafica.
Si consideri la realtà di riferimento sopra descritta e si realizzino:

1. la progettazione concettuale della realtà indicata attraverso la produzione di uno schema (ad esempio ER, Entity-Relationship) con gli attributi di ogni entità, il tipo di ogni relazione e i suoi eventuali attributi;
2. una traduzione dello schema concettuale realizzato in uno schema logico (ad esempio secondo uno schema relazionale);
3. le seguenti interrogazioni espresse in algebra relazionale e/o in linguaggio SQL:
    a. produrre l elenco dei soci (con cognome, nome e telefono) che hanno un debito nella BdT (coloro che hanno usufruito di ore di prestazioni in numero superiore a quelle erogate);
    b. data una richiesta di prestazione, visualizzare la porzione di mappa del territorio nel quale si trova il socio richiedente e l elenco di tutti i soci che si trovano in quella zona in grado di erogare quella prestazione, visualizzandone il nome, cognome, indirizzo e numero di telefono;
    c. visualizzare tutti i soci che fanno parte della segreteria e che offrono anche altri tipi di prestazione;
    d. produrre un elenco delle prestazioni ordinato in modo decrescente secondo il numero di ore erogate per ciascuna prestazione
4. (Facoltativo) Sviluppare il problema posto scegliendo una delle due seguenti proposte descrivendone le problematiche e le soluzioni tecniche adottabili:
    4.1 L'associazione BdT vuole realizzare un sito Web per rendere pubbliche le sue attività consentendo anche di effettuare on-line le interrogazioni della base di dati previste nel punto 3;
    4.2 L'associazione BdT vuole realizzare un sito Web attraverso il quale possa raccogliere l'adesione on-line di altri associati, attraverso il riempimento di un modulo da inviare via Internet all associazione

Analisi

Per l'analisi della realtà di interesse si utilizza la modellazione Entità-Relazioni che produce un diagramma ER.

Individuazione delle entità.

Entità 'socio': è l'anagrafica dei soci. Ogni socio deve essere registrato come una singola istanza di questa entità per potere operare nella BDT. La chiave primaria è artificiale, numerica ad autoincremento. Gli attributi essenziali sono 'cognome', 'nome', 'indirizzo', telefono' tutti di testo.

Entità 'attività': è il catalogo delle prestazioni realizzabili nell'ambito della BDT. Ogni tipo di attività fornibile deve essere registrata come una singola istanza di questa entità per potere essere fornita nell'ambito della BDT. Ogni socio deve offrire la disponibilità ad effettuare per gli altri soci una o più attività nell'ambito di questo catalogo. La chiave primaria è artificiale, numerica ad autoincremento. L'attributo essenziale è la 'descrizione' dell'attività di tipo testo.

Entità 'categoria': è una tabella di look-up che raggruppa le attività affini. Ad esempio nella categoria 'lavori manuali' saranno raggruppati 'muratore','idraulico','piastrellista', ecc ... La chiave primaria è artificiale, numerica ad autoincremento. L'attributo essenziale è la 'descrizione' della categoria di tipo testo.

Entità 'zona': è la descrizione della struttura dell'area servita dalla BDT. Ogni zona dell'area servita è rappresentata da una istanza. La chiave primaria è artificiale, numerica ad autoincremento. Gli attributi essenziali sono la 'descrizione' di tipo testo e l'immagine' di tipo binario.

Entità 'mappa': non è una vera è propria entità in quanto contiene una unica istanza priva di chiave primaria. Questa una istanza contiene un attributo di tipo testo che contiene il nome della BDT e un attributo di tipo binario che consente di ospitare la mappa dell'intera area servita dalla BDT

Individuazione delle associazioni.

Associazione socio/zona 'abita': ogni socio deve abitare in una zona dell'area servita dalla BDT mentre in ogni zona ci possono essere più soci. Esiste quindi una associazione 1:N tra zona e socio. La relazione è totale dal lato socio (un socio deve inserire l'appartenenza alla zona al momento dell'iscrizione) e parziale dal lato zona (una zona può temporaneamente essere priva di soci)

Associazione socio/attività 'disponibile': ogni socio deve offrire una o più disponibilità ad effettuare attività e una attività può essere attuata da più di un socio. Esiste quindi una associazione N:N tra attività e socio. La relazione è totale dal lato socio (un socio deve inserire almeno una disponibilità al momento dell'iscrizione) e parziale dal lato attività (una attività può temporaneamente essere priva di soci in grado di realizzarla)

Associazione attività/categoria 'appartiene': ogni attività deve appartenere ad una categoria mentre ogni categoria può possedere più attività. Esiste quindi una associazione 1:N tra categoria e attività. La relazione è totale dal lato attività (una attività deve essere assegnata ad una categoria al momento dell'inserimento) e totale dal lato categoria (una categoria deve avere attività)

Entità debole 'prestazione': un socio dopo avere consultato l'elenco delle disponibilità, anche in relazione alla zona può chiedere una prestazione ad un altro socio che la effettua. Si ipotizza quindi l'esistenza di una entità debole che rappresenta lo storico delle prestazione effettuate. La chiave primaria è artificiale, numerica ad autoincremento. Gli attributi essenziali sono la 'data' di effettuazione di tipo data e la quantità di ore impiegate di tipo numerico. L'esistenza di

una istanza di questa entità è determinata da due associazioni a due diverse istanze della stessa entità socio che individuano il prestatore ed il ricevente. In questo modo si evitano ridondanze nella definizione della prestazione che è registrata in un punto solo.

Associazione socio/prestazione #1 'fornisce': ogni prestazione deve essere associata ad un socio che la fornisce mentre un socio può fornire più prestazioni (in tempi diversi). Esiste quindi una associazione 1:N tra socio e prestazione. La relazione è totale dal lato prestazione (una prestazione deve essere associata ad un fornitore) e parziale dal lato socio (un socio può non avere mai fornito nulla)

Associazione socio/prestazione #2 'riceve': ogni prestazione deve essere associata ad un socio che la riceve (ad eccezione delle prestazioni di segreteria) mentre un socio può ricevere più prestazioni (anche allo stesso tempo). Esiste quindi una associazione 1:N tra socio e ricevente. La relazione è parziale dal lato ricevente (le prestazioni di segreteria non hanno riceventi) e parziale dal lato socio (un socio può non avere mai ricevuto nulla)

Associazione prestazione/attività 'ditipo': ogni prestazione deve essere associata ad una attività mentre una attività può essere associata a più prestazioni. Esiste quindi una associazione 1:N tra attivita e prestazione. La relazione è totale dal lato prestazione (una prestazione deve essere associata ad una attività) e parziale dal lato attività (una attività può non essere mai stata prestata)

Ipotesi aggiuntive

Gestione della grafica: L'entità mappa serve per dare generalità alla soluzione del problema consentendo una agevole personalizzazione della BDT in realtà diverse. L'immagine raster della mappa è memorizzata nell'unica istanza di tale entità e può essere sostituita mediante l'interfaccia della banca dati per adattare la BDT ad altre realtà. In questo modo si ottiene un sistema flessibile nel quale l'intefaccia non è in alcun modo legato alla realtà di interesse.
L'immagine viene presentata nell'home page del sito associata ad una mappa html client-side (tag IMG/MAP/AREA) generata a partire dall'attributo 'cl_area' dell'entità 'zona' di tipo testo che contiene la definizione dell'area poligonale nella forma: "x1,y1,x2,y2 ..."

Tag HTML di presentazione della mappa principale :

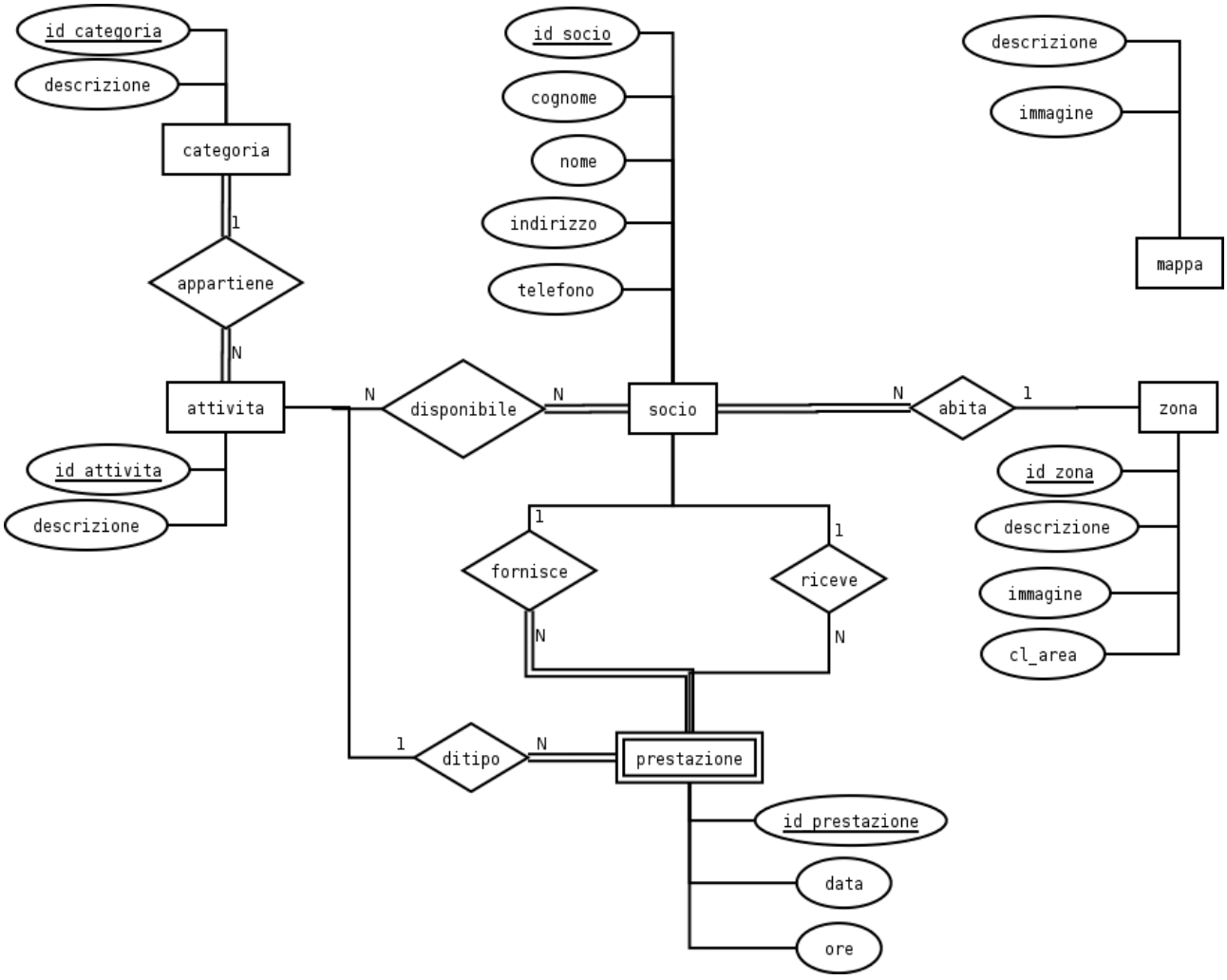
```
<IMG SRC="url-immagine" USEMAP="#mappa">
<MAP NAME="mappa" >
<AREA HREF="mostra_zona.php?id_zona=1" SHAPE="poly" COORDS= "x1,y1,x2,y2 ..." >
...
<AREA HREF=" mostra_zona.php?id_zona=n" SHAPE="poly" COORDS= "x1,y1,x2,y2 ..." >
```

Gli elementi in corsivo sono generati dinamicamente in base a dati archiviati in banca dati.
I collegamenti mandano a pagine secondarie che mostrano il dettaglio di zona e gli utenti presenti nella zona.

La scelta dei campi binari (BLOB) per l'archiviazione delle immagine ha il vantaggio di mantenere tutte le informazioni archiviate attraverso il DMBS facilitando le operazioni di backup e rendendo il sistema totalmente indipendente dal file-system del sistema ospite.
D'altra parte questa scelta rende più complessa l'interfaccia che deve generare dinamicamente le immagini al momento della presentazione. La scelta alternativa di archiviare per le immagini solo i percorsi nel file-system avrebbe reso più semplice la gestione dell'interfaccia ma avrebbe spezzato l'archivio del sistema informativo in due parti una gestita dal DBMS e l'altra dal file-system del sistema ospite.
Gestione della segreteria: Le attività di segreteria svolte da un prestatore non possono essere direttamente associate ad un ricevente. Si ipotizza che le attività di segreteria non abbiano ricevente (attributo nullo). In questo modo il bilancio complessivo della banca non è in pareggio ma è sempre leggermente in perdita a causa dell'attività dei segretari.
Una ipotesi alternativa è di considerare che le attività di segreteria si svolgano solo in relazion
a bisogni di altri soci e quindi la segreteria torna ad essere una normale attività.

Schema concettuale

Si usa la tecnica dello schema ER
Il rettangolo rappresenta le entità
Il rettangolo doppio rappresenta le entità deboli
L'ovale rappresenta gli attributi delle entità
Il testo sottolineato individua la chiave primaria
Il rombo rappresenta le associazioni
I collegamenti doppi rappresentano il legami totali
I collegamenti singoli rappresentano il legami parziali



Schema logico

Si usa il modello relazionale.
Ogni entità ed entità debole viene sostituita con una relazione.
Ogni associazione 1:N tra due entità viene sostituita con l'esportazione della chiave primaria del lato 1 come chiave esterna nel lato N.
Ogni associazione N:N tra due entità viene sostituita con la generazione di una nuova relazione ottenuta esportando le due chiavi primarie come chiavi esterne e definendo la loro composizione come chiave primaria della nuova relazione.
Poiché il progetto non richiede il modello fisico nello schema logico vengono anche indicati i tipi di dato.

Relazioni

socio(id_socio:intero, cognome:testo, nome:testo, indirizzo:testo, tel:testo, id_zona:intero)

attivita(id_attivita:intero, descrizione:testo, id_categoria:intero)

categoria(id_categoria:intero, descrizione:testo)

zona(id_zona:intero, descrizione:testo, immagine:binario, cl_area:testo)

mappa(descrizione:testo, immagine:binario)

disponibile(id_socio:intero,id_attivita:intero)

prestazione(id_prest:intero, data:data, ore:intero, id_forn:intero, id_ric:intero, id_att:intero)

Vincoli di integrità

Vincoli di integrità referenziale

socio.id_zona $\subseteq$ zona.id_zona
attivita.id_categoria $\subseteq$ categoria.id_categoria
disponibile.id_socio $\subseteq$ socio.id_socio
disponibile.id_attivita $\subseteq$ attivita.id_attivita
prestazione.id_fornitore $\subseteq$ socio.id_socio
prestazione.id_ricevente $\subseteq$ socio.id_socio | $\varnothing$
prestazione.id_attivita $\subseteq$ attivita.id_attivita

Vincoli espliciti

Tutti i campi non nulli eccetto id_ricevente

prestazione.ore > 0

Interrogazioni

a. elenco dei soci che hanno un debito nella BdT

Per evitare ridondanze non sono archiviati per ogni socio né il saldo, né credito e debito che devono quindi essere calcolati in base alle prestazioni fornite e ricevute.
Ogni istanza di socio è associata doppiamente alle istanze di prestazione attraverso le associazioni 'fornisce' e 'riceve'. Per determinare credito e debito di ciascun socio si realizzano due natural join tra socio e prestazione attraverso queste due associazioni; entrambe vengono raggruppate per 'id_socio' e sommate per 'ore'; in questo modo la prima restituisce un elenco dei soci con la somma delle ore in credito mentre la secondo un analogo elenco con la somma delle ore in debito.
Il risultato di queste due giunzioni viene messo in left join con l'entità socio ottenendo un elenco di **TUTTI** i soci con rispettivo credito e debito. Poiché un socio potrebbe non avere fatto alcuna prestazione e/o alcuna richiesta la sua istanza potrebbe non esistere in uno o entrambi i join interni quindi il left join esterno restituisce NULL per il suo credito e/o debito; non si possono ignorare tali soci perché se credito è NULL ma debito esiste il socio è "in debito". Il saldo è un campo calcolato come differenza tra credito e debito ma poiché le elaborazioni algebriche contenenti un NULL producono sempre NULL è necessario convertire i NULL in 0 con la funzione IFNULL(exp1,exp2) che restituisce exp1 se exp1 non è NULL altrimenti restituisce exp2. La restrizione rimuove dall'elenco tutti i soci che non hanno debito.

```
SELECT
     t1.cognome,
     t1.nome,
     IFNULL(tc.credito,0) - IFNULL(td.debito,0) AS saldo
FROM
     socio AS t1
     LEFT JOIN (
          SELECT
             t2.id_socio,
             SUM(t3.ore) AS credito
          FROM
             socio AS t2,
             prestazione AS t3
          WHERE
              t2.id_socio=t3.id_fornitore
          GROUP BY t2.id_socio
          ) AS tc ON t1.id_socio=tc.id_socio
     LEFT JOIN (
          SELECT
             t4.id_socio,
             SUM(t5.ore) AS debito
          FROM
             socio AS t4,
             prestazione AS t5
          WHERE
             t4.id_socio=t5.id_ricevente
          GROUP BY t4.id_socio
          ) AS td ON t1.id_socio=td.id_socio
WHERE
   IFNULL(tc.credito,0) - IFNULL(td.debito,0) < 0
   ORDER BY
     t1.cognome,t1.nome
```

b. mappa della zona del richiedente e soci della stessa zona in gradi di erogare la prestazione

Si deve prima di tutto determinare la zona del socio richiedente. Supponendo che a causa dell'autenticazione in fase di accesso sia già disponibile la chiave primaria del socio ($id_socio) si estrae la chiave di zona (id_zona) con una restrizione sulla tabella 'zona' (o sulla tabella 'socio'). Se l'autenticazione iniziale ha reso già disponibile questo dato la select annidata non è necessaria.

Supponendo di conoscere la richiesta in base alla sua descrizione ($request) si effettua un natural join tra socio, disponibile, attività e zona con le seguenti restrizioni:

- Zona del socio richiedente (già disponibile o determinata dalla select annidata)
- Attività corrispondente alla descrizione (già disponibile)
- Socio diverso dal richiedente (non è essenziale ma elimina dati ovvi)

La proiezione restituisce l'immagine della porzione di mappa legata alla zona e i dati del fornitore. L'immagine risulta quindi ripetuta in ogni istanza; sarà l'interfaccia che provvederà ad utilizzarla solo una volta per la presentazione ignorando tutte le estrazioni successive.

In alternativa si potevano fare due query.

```
SELECT
     t4.immagine,
     t1.cognome,
     t1.nome,
     t1.indirizzo,
     t1.telefono
FROM
     socio AS t1,
     disponibile as t2,
     attivita as t3,
     zona as t4,
WHERE
     t1.id_socio=t2.id_socio
AND
     t2.id_attivita=t3.id_attivita
AND
     t4.id_zona=t1.id_zona
AND
     t4.id_zona=SELECT
                    t5.id_zona
                FROM
                    zona as t5
                WHERE
                    t5.id_socio=$id_user
AND
     t3.descrizione='$request'
AND
     t1.id_socio<>$id_user
```

c. soci che fanno parte della segreteria e che offrono anche altri tipi di prestazione;
Prima di tutto si estraggono i soci che sono disponibili per attività diverse da 'segeteria'.
La proiezione estrae solo id_socio e forza l'unicità della riga.
La giunzione naturale lega la tabella attività con l'associazione disposizione;la tabella socio non serve a questo livello perchè l'anagrafica del socio non serve nella query interna.
La restrizione esclude le righe che si riferiscono a 'segreteria'
Il risultato è una tabella virtuale del tipo t_att-seg(id_socio) e contiene gli id di tutti i soci che hanno attività diversa da segreteria inseriti una volta sola

Questa tabella viene inserita in una query che effettua una giunzione naturale tra attività e socio attraverso l'associazione 'disponibile'; la proiezione estrae id_socio, cognome e nome.
La restrizione è composta da due parti:

- la prima impone che l'attività sia di tipo 'segreteria'
- la seconda impone che l'id_socio si contenuto (operatore IN) anche nella tabella risultante della query annidata.

In questo modo la query esterna restituisce i soci che sono disponibili a fare 'segreteria' e sono disponibili anche a fare almeno un'altra attività.

```
SELECT
   t1.cognome,
   t1.nome
FROM
   socio as t1,
   disponibile as t2,
   attivita as t3
WHERE
   t1.id_socio=t2.id_socio
AND
   t2.id_attivita=t3.id_attivita
AND
   t3.descrizione='segreteria'
AND
   t1.id_socio IN (SELECT
                       t4.id_socio
                   FROM
                       disponibile as t4,
                       attivita as t5
                   WHERE
                       t4.id_attivita=t5.id_attivita
                   AND
                       t5.descrizione<>'segreteria'
                  )
```

d. elenco delle prestazioni con ordinamento decrescente del numero di ore erogate

Si effettua una giunzione naturale tra prestazione e attività che determina un elenco delle prestazioni effettivamente erogato con la descrizione della attività associata.
Si effettua un raggruppamento per id_attività ed una somma delle ore di ciascun tipo di attività. La proiezione contiene la descrizione dell'attività ed il totale delle ore associate, l'ordinamento è decrescente secondo questo totale.

```
SELECT
     t2.descrizione,
     SUM(t1.ore) AS totale
FROM
     prestazione AS t1,
     attivita AS t2
WHERE
     t1.id_attivita=t2.id_attivita
GROUP BY t2.id_attivita
ORDER BY totale DESC
```

Interfaccia

4.2 Modulo di iscrizione online

Il modulo di iscrizione online si compone di una pagina di richiesta e di una pagina di risposta.
La pagina di richiesta contiene un modulo HTML (tag <FORM>) che contiene i campi moduli compilabili da parte dell'utente in forma di testo (tag <INPUT>) per cognome, nome, indirizzo, telefono e in forma di selezione (tag <SELECT>) semplice per zona e multipla per attività. La pagina invia i dati con il metodo POST.
La pagina di risposta è uno script che verifica l'esistenza e la correttezza dei valori inseriti nei campi moduli e ricevuti attraverso un POST, li inserisce in banca dati ed effettua una eco di conferma.

Si ipotizza di usare una piattaforma xAMP composta da:

- x sistema operativo Windows o Linux
- A web server Apache
- M sql server MySQL
- P script interprete PHP

Parti comuni ad entrambe le pagine

Entrambe le pagine devono effettuare una connessione alla banca dati quindi lo script di connessione viene realizzato come un file di inclusione denominato connect.php.
Lo script di inclusione rende disponibile una connessione tcp (handle $sock) verso il DBMS locale e seleziona la banca dati dell'applicazione (bdt).
Se la connessione ha successo lo script è silente mentre se fallisce la pagina termina prematuramente con la segnalazione di errore.

```
<?php
 //connect.php
 $sock=mysql_connect("localhost","nobody","");
 if ($sock==0) die(mysql_error());
 $ris=mysql_select_db("bdt",$sock);
 if ($ris==0) die(mysql_error());
?>
```

Pagina di richiesta
La pagina di richiesta contiene il modulo HTML ed i relativi campi modulo.
I campi di testo sono statici mentre i campi di selezione devono essere popolati dinamicamente estraendo i dati dalla banca dati.
E quindi necessario effettuare una connessione usando lo script di connessione precedentemente definito.
Ottenuta una connessione si estraggono le due query che servono per popolare rispettivamente i campi modulo di zona e di attività ottenendo i due handle relativi ($qzona e $qattivita).
La parte HTML della pagina contiene un modulo che invia con il metodo POST i dati inseriti localmente alla pagina di risposta risposta.php.
I campi modulo relativi a nome, cognome, indirizzo e telefono sono di tipo "text".
Il campo modulo relativo alla zona è di tipo "select" a scelta singola e viene popolato con gli elementi estratti dalla query $qzona.
Il campo modulo relativo alla zona è di tipo "select" a scelta multipla e viene popolato con gli elementi estratti dalla query $qattivita. Un campo select a scelta multipla può avere più selezioni contemporaneamente (combinazioni CAP+click e CTRL+click) e la sua variabile locale va definita come array in modo da trasmettere via POST i valori multipli.
Tutti i campi moduli hanno nomi che corrispondono ai nomi degli attributi in banca dati eccetto il campo delle attività che è un array.
Per brevità sono state omesse tutte le parti non essenziali del codice HTML.

```
<?php
 require "connect.php";
 $msg="SELECT id_zona,descrizione FROM zona";
 $qzona=mysql_query($msg,$sock);
 $msg="SELECT id_attivita,descrizione FROM attivita";
 $qattivita=mysql_query($msg,$sock);
?>
<form action="risposta.php" method="POST">
 Inserire il cognome:
 <input type="text" name="cognome">
 ... altri campi testo
 Selezionare una zona:
 <select name="id_zona">
 <?php while($riga=mysql_fetch_assoc($qzona) { ?>
 <option value="<?php echo $riga["id_zona"]?>">
  <?php echo $riga["descrizione"]?>
 </option>
 </select>
 <?php } ?>
 Selezionare una o più attività:
 <select name="att_array[]" size="10" multiple>
 <?php while($riga=mysql_fetch_assoc($qattivita) { ?>
 <option value="<?php echo $riga["id_attivita"]?>">
  <?php echo $riga["descrizione"]?>
 </option>
 </select>
 <?php } ?>
 <input type="submit" value="invia">
</form>
```

Pagina di risposta
La pagina di risposta riceve i campi modulo nell'array $_POST[].
E necessario effettuare una connessione usando lo script di connessione precedentemente definito.
La prima operazione è verificare l'esistenza e la correttezza dei valori comunicati.
In caso di errore la pagina termina prematuramente con una segnalazione altrimenti vengono fatti gli inserimenti in banca dati della nuova istanza di socio e delle nuove istanze di associazione 'disponibile' tra socio ed attività.
La chiave primaria del nuovo socio è generata per autoincremento quindi per potere inserire le istanze di associazione in 'disponibile' è necessario estrarre la chiave primaria che a priori è sconosciuta.
Se gli inserimenti falliscono la pagina termina prematuramente con una segnalazione di errore altrimenti viene fatta una eco dei dati inseriti.
Il campo modulo id_attivita, essendo multiplo, è contenuto in un array di dimensione non determinate a priori. E' quindi necessario percorre l'intero array per estrarre i valori di id_attivita selezionati dall'utente.
Delle disponibilità viene comunicata solo la chiave quindi per mostrare la descrizione è necessario interrogare la banca dati.
Per brevità sono state omesse tutte le parti non essenziali del codice HTML.

```
<?php
 require "connect.php";
 if (isset($_POST["cognome"])) $cognome=$_POST["cognome"];
 else die("manca cognome");
 ...
 if (isset($_POST["att_array"])) $att_array=$_POST["att_array"];
 else die("manca disponibilità");
 $msg="INSERT socio
        (cognome,   nome,   indirizzo,telefono,id_zona)
       VALUES(
        '$cognome','$nome','$indirizzo','$telefono','$id_zona')"
 $qsocio=mysql_query($msg,$sock);
 if ($qsocio==0)die(mysql_error());
 $id_socio=mysql_insert_id($sock);
 echo "id_socio: $id_socio<br>";
 echo "cognome: $cognome<br>";
 ...
 foreach($att_array as $idattivita) {
  $msg="SELECT
          descrizione
        FROM
          attivita
        WHERE
         id_attivita=$id_attivita"
  $qatt=mysql_query($msg,$sock);
  $attivita=mysql_fetch_assoc($qatt);
  echo "disponibile per: ".$attivita["descrizione"]. "<br>";
  $msg="INSERT disponibile
         (id_socio,id_attivita)
        VALUES($id_socio,$id_attivita)"
  $qatt=mysql_query($msg,$sock);
 }
?>
```

Esempio di presentazione delle pagine descritte

Pagina di richiesta

inserire il cognome:

inserire il nome:

inserire l'indirizzo:

inserire il telefono:

Selezionare una zona: Santa viola

Selezionare una o più attivita:

biciclette
muratore
informatica
torte

invia

Pagina di risposta

cognome: Rossi
nome: Mario
indirizzo: via Roma,1
telefono: 051123456
zona: San Vitale
disponibile per: torte
disponibile per: informatica
disponibile per: biciclette