Database di supporto al sito WEB dell'ISTI. Import globale da WebServices dei dati relativi al personale.

Renzo Beltrame, Federico Ponchio, Claudia Raviolo
Istituto di Scienza e Tecnologie dell'Informazione "Alessandro Faedo"

Nota Interna dicembre 2015

Sommario: La Nota illustra brevemente la procedura di import globale dei dati che il sito Web eredita dal database del personale.
Abstract: The ISTI website uses a database to store information presented on its pages. We sketch here the updated structure of this database.

La nota si aggiunge alla nota interna 2012-B4-03 che descrive il database di supporto al sito WEB dell'Istituto.

Ricordiamo che questo database è implementato in *Postgres* e per l'immissione dei dati impiega il backend offerto dal sistema *Symfony* (nella versione 1.4) con minime personalizzazioni. Il frontend è invece costituito interamente da procedure .php scritte ad hoc su uno scheletro di classi che ne assicurano una struttura ripetitiva e una logica unitaria. Le parti in javascript sono appoggiate a librerie (jquery, mootools) distribuite free.

Nell'impostazione del database l'organizzazione dell'Istituto è stata vista come l'articolazione dell'attività svolta stabilmente, o comunque di lungo periodo. A cui è aggiunta l'attività non continuativa: workshop, seminari, corsi, etc..
Alle singole attività sono correlate le persone che le svolgono. Le informazioni sulle persone sono ridotte a quelle strettamente legate a queste funzioni, e sono importate dal sistema informativo di Istituto con cadenza giornaliera. Sul db del sito non vi sono quindi informazioni sensibili relative alle persone.
Vi sono due ulteriori raggruppamenti logici, uno relativo alla logistica, e l'altro ai documenti - verbali, normative, modulistica, etc. Questi ultimi, infatti, possono comparire in più pagine del sito e con funzioni diverse.

Il programma effettua un aggiornamento globale sulla base di WebServices che mettono a disposizione soltanto i dati relativi al personale che sono di interesse per il sito WEB.

La nota è costituita dal sorgente del programma di import, che è ragionevolmente autodocumentato da commenti inseriti nei vari punti, e da un esempio del suo uso.

Il programma che importa i dati.

Il programma che importa i dati si compone di un programma principale e di una raccolta di funzioni usate ripetutamente nel programma principale che sono raccoolte nel file util.php.

Il programma principale.

```
<?php
// versione gennaio 2015 [r. beltrame]
/* Aggiunge id_si per scambio con Sistema Informativo
  Aggiorna la table persone la prima volta
  ricordarsi cod_fiscale (chiave) -> 16 caratteri.
*/
    require_once('util.php');
    global $db;

// authorization
    echo("<? require_once('/var/www/include/authadm.php'); ?> \n");

    echo(" <html> \n <!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.01 Transitional//EN\" .
    "\"http://www.w3.org/TR/html4/loose.dtd\"> \n " ) ;
    echo(" <body> \n " ) ;
```

```
    $ts = date(" Y-m-d H:i:s");
    echo("<h4>Dbname: isti{utf8 - Importing PERSONALE data from webservices on $ts </h4>
     \n");
    echo("(i dati del Direttore vanno inseriti e corretti direttamente sul db) <br />
    <br /> \n\n");
    echo("<div style=\"font-size: 14px\"; \"line-height: 1.5\" > \n <p> " ) ;

/* // put 0 on id_si Null
    $query = "UPDATE persone SET id_si = 0 WHERE id_si IS NULL ;" ;
    executeQuery($db, $query);
    exit;
*/
/*
// insert id_si da PERSONALE
   IdSiPersons();
   exit;
*/

    $ret1 = updatePersonale('Step_1'); // dati anagrafici da elenco

    $ret2 = updatePersonale('Step_2'); // dati da schede singole persone

    checkAttiviAll();
    checkWebtitlesAll();

    echo(" </p> \n </div> \n </body> \n </html> \n" );
    exit;

/* ======== Functions ==================== Functions=========*/

/* Master function updatePersonale
  I dati del direttore vanno inseriti e corretti direttamente sul db del sito WEB */

/* Structure of a $elenco
Array
(
    [id] => 170
    [nome] => Nicola
    [secondoNome] =>
    [cognome] => Aloia
    [codiceFiscale] => LAONCL52L30F784C
    [profilo] => Collaboratore Tecnico Enti di Ricerca
    [subprofilo] =>
    [al] =>
    [english] => Technical Staff
    [gruppi] => Array
        (
            [0] => NeMIS
        )

)

And for single person

Array
(
    [id] => 170
    [nome] => Nicola
    [secondoNome] =>
    [cognome] => Aloia
    [profilo] => Collaboratore Tecnico Enti di Ricerca
    [subprofilo] =>
    [al] =>
    [english] => Technical Staff
    [email] => nome.cognome@isti.cnr.it
    [gruppi] => Array
        (
            [0] => NeMIS
        )

    [stanze] => Array
        (
            [0] => Array
                (
                    [stanza] => C-37
                    [ingresso] => 20
                    [piano] => 1P
                    [edificio] => B
                )

        )

    [recapitiTelefonici] => Array
        (
            [0] => Array
                (
                    [tipoRecapitoTelefonico] => Ufficio
                    [numero] => +39 050 3152997
                )
```

```
            [1] => Array
                (
                    [tipoRecapitoTelefonico] => Mobile aziendale
                    [numero] => +39 348 3966801
                )

        )

)
*/

 function updatePersonale($step){
        global $db, $ucodes;

    $wsdl_url = "http://pimpa.isti.cnr.it/PERSONALE/web-services/people/people.wsdl";
    ini_set("soap.wsdl_cache_enabled", "0");
    $client = new SoapClient($wsdl_url);
    $quit = false;

    $elenco_object = $client->personale();
    if($step == 'Step_1'){
            foreach($elenco_object as $persona){
                $query = "UPDATE persone SET webtitle = '{$persona->english}' ".
                "WHERE id_si = {$persona->id}";
//                  echo $query."<br/>";
                    executeQuery($db, $query);
            }
    }
    $elenco = object_to_array_recusive($elenco_object);  // array of arrays
   // preint_r($elenco);

    $query ="SELECT * FROM persone WHERE interno = TRUE ;";
    $attivi = getRecords($db,$query);
    $query ="SELECT * FROM persone WHERE id_si > 0 ;";    //  id_si diverso da 0 !!
    $registrati = getRecords($db,$query);

    $idse = arrayColumn($elenco, 'id');
    $idsa = arrayColumn($attivi,'id_si');
    $idrg = arrayColumn($registrati,'id_si');
//    preint_r($idsa);

    $idadd = array_diff($idse,$idrg);  // to be added
    $idract = array_diff($idse,$idsa,$idadd); // to be reactivated
    $idcanc = array_diff($idsa,$idse);  // to be cancelled
    $idcheck = array_intersect($idse,$idsa);  // to be checked

    $de = count($elenco);
    $da = count($attivi);
    $dr = count($registrati);
    $a1 = count($idse);
    $a2 = count($idsa);
    $a3 = count($idrg);
    $c1 = count($idcanc);
    $c2 = count($idadd);
    $c3 = count($idract);
    $c4 = count($idcheck);
    echo("Counts $step - elenco: $de $a1  attivi: $da  $a2  ".
        "registrati: $dr $a3  canc: $c1  add: $c2  ".
        "react: $c3  check: $c4  <br /> \n \n");
//    preint_r($idcanc);
//      $i=42; preint_r($elenco[$i]);

    $ucodes = getUnits();  // codici => id_units attive
//  preint_r($ucodes);
    $ret = 0;

      if($c1 >0) $ret += cancelPersons($idcanc);  // cancel retired
      if($c2 >0) $ret += newPersons($idadd,$elenco);  // add new persons
      if($c3 >0) $ret += reactivatePersons($idract,$elenco);  // reactivated retired

    if($step === "Step_2"){
      if($c4 >0) $ret += checkPersons($idcheck,$elenco);  // check persons
    }
    countUpdate($elenco);
    return $ret;
 }
//  END function updatePersonale

/* Count of the updates */
function countUpdate($elenco)
{
$query ="SELECT * FROM persone WHERE interno = TRUE ;";
    $attivi = getRecords($db,$query);
    $query ="SELECT * FROM persone WHERE id_si > 0 ;";    //  id_si diverso da 0 !!
    $registrati = getRecords($db,$query);

    $idse = arrayColumn($elenco, 'id');
    $idsa = arrayColumn($attivi,'id_si');
    $idrg = arrayColumn($registrati,'id_si');
//    preint_r($idsa);
```

```
    $idadd = array_diff($idse,$idrg);  // to be added
    $idract = array_diff($idse,$idsa,$idadd); // to be reactivated
    $idcanc = array_diff($idsa,$idse);  // to be cancelled
    $idcheck = array_intersect($idse,$idsa);  // to be checked

    $de = count($elenco);
    $da = count($attivi);
    $dr = count($registrati);
    $a1 = count($idse);
    $a2 = count($idsa);
    $a3 = count($idrg);
    $c1 = count($idcanc);
    $c2 = count($idadd);
    $c3 = count($idract);
    $c4 = count($idcheck);
    echo("Counts $step - elenco: $de $a1  attivi: $da  $a2  ".
    "registrati: $dr $a3  canc: $c1  add: $c2  react: $c3  ".
    "check: $c4  <br /><br /> \n \n");
}

function cancelPersons($idcanc)
{
/* Cancel retired , i.e.  set interno =FALSE  and fine = CURRENT_DATE on all his activities
  I dati del direttore vanno inseriti e corretti direttamente sul db del sito WEB */
  global $db;

  $quit = 0;
  foreach($idcanc as $id) {
      $id_p = idsi_to_idpers_uniq($id);
      if($id_p == 0){
         echo("id_si = $id non univoco ( id_persona = $id_p ) <br /> \n");
         $quit += 1;
      } else
      {
         $query = "SELECT nome, cognome FROM persone WHERE id_persona = '$id_p' AND ".
          "id_si = '$id' ;";
         $anag = getRecords($db,$query);
//       preint_r($anag);
         $query = "UPDATE persone SET interno = FALSE WHERE id_persona = '$id_p' ".
          "AND id_si = '$id' ;";
         executeQuery($db, $query);
        $query = "UPDATE attivita SET fine = CURRENT_DATE WHERE id_persona = '$id_p' ;";
        executeQuery($db, $query);
        echo("Retired {$anag[0]['nome']} {$anag[0]['cognome']} id_si = $id <br /> \n");
      }
  }
  return $quit;
}

/* Reactivate registered persons */
function reactivatePersons($idract,$elenco){
  global $db, $ucodes;

  $quit = 0;
  foreach($idract as $id) {
      $id_p = idsi_to_idpers_uniq($id);   // preint_r($idract, $id, $id_p);
      if($id_p == 0){
         echo("id_si = $id non univoco ( id_persona = $id_p )  <br /> \n");
         $quit += 1;
      } else
      {
         $query = "SELECT nome, cognome FROM persone WHERE id_persona = '$id_p'
         AND id_si = '$id' ;";
         $anag = getRecords($db,$query);  // preint_r($query,$anag);
         $query = "UPDATE persone SET interno = TRUE  WHERE id_persona = '$id_p'
         AND id_si = '$id' ;";
         executeQuery($db, $query);
         echo("Reactivated {$anag[0]['nome']} {$anag[0]['cognome']} id_si = $id <br /> \n");
      }
  }
  return $quit;
}

/* Insert new persons *
  I dati del direttore vanno inseriti e corretti direttamente sul db del sito WEB */
function newPersons($idadd, $elenco){
  global $db, $ucodes;

  $quit = 0;
  foreach($idadd as $id) {
      $id_p = idsi_to_idpers_uniq($id); // preint_r($idadd, $id, $id_p);
      if($id_p !== 0){
         echo("id_si = $id non univoco in insert new person <br /> \n");
         $quit += 1;
      } else
      {
          $kid = array_keys($idadd, $id); // preint_r($kid);
          if(count($kid) != 1){
            echo("id_si = $id non univoco  in selezione elenco <br /> \n");
```

```
            $quit += 1;
          } else
          {
              $anap = $elenco[$kid[0]];
              array_splice(&$anap, 5);
              $anap = change_key($anap, "id", "id_si");
              $anap = change_key($anap, "secondoNome", "secondo_nome");
              $anap = change_key($anap, "codiceFiscale", "chiave");
              $query = db_mount_insert("persone", $anap);  // preint_r($ana); preint_r($query);
              executeQuery($db, $query);
              echo("Inserted {$anap['nome']} {$anap['cognome']} id_si = $id <br /> \n");
//        set interno = true
              $id_p = idsi_to_idpers_uniq($id);
              if($id_p == 0){      // if id_persona inserito
                  echo("Incorrect insertion of a new person {$anap['nome']} {$anap['cognome']}".
                  " id_si = $id  <br /> \n");
                  $quit += 1;
              } else
              {
                  $query = "UPDATE persone SET interno = TRUE  WHERE id_persona = '$id_p' ".
                  "AND id_si = '$id' ;";
                  executeQuery($db, $query);
                  echo("Activated {$anap['nome']} {$anap['cognome']} id_si = $id  -  ".
                  "Check webtitle <br /> \n");
              }
          }
      }
  }
  return $quit;
}
/* Update affiliazioni  e altri dati  delle persone
  I dati del direttore e del Consiglio di Istituto vanno
  inseriti e corretti direttamente sul db del sito WEB */
function checkPersons($idcheck, $elenco){
  global $db, $ucodes;

  $k = 0;
  $quit = 0;
  foreach($idcheck as $id) {
      $id_p = idsi_to_idpers_uniq($id); // var_dump($id, $id_p);
      if($id_p == 0){
         echo("id_si = $id non univoco in check person <br /> \n");
         $quit += 1;
      } else
      {
          $query = "SELECT nome, cognome, chiave, interno FROM persone ".
                  "WHERE id_persona = '$id_p' AND id_si = '$id' ;";
          $anag = getRecords($db,$query);  //
//        $nanag = pg_num_rows($anag);  // different results (null)
          $nanag = count($anag);
          if($nanag != 1){
              echo("Not unique in dbisti {$anag[0]['nome']} {$anag[0]['cognome']} ".
              "id_si = $id records: $nanag <br /> \n");
              print_r($anag);
          }
          else
          {
            if(!$anag[0]['interno']){
                $query = "UPDATE persone SET interno = TRUE  WHERE id_persona = '$id_p' ".
                    "AND id_si = '$id' ;";
                executeQuery($db, $query);
                echo("Reactivated !? {$anag[0]['nome']} {$anag[0]['cognome']} ".
                "id_si = $id <br /> \n");
            }
            $kid = array_keys($idcheck, $id); // preint_r($kid);
            if(count($kid) != 1){
              echo("id_si = $id non univoco  in selezione elenco <br /> \n");
              $quit += 1;
            }
            else
            {
              $el = $elenco[$kid[0]]; //preint_r($el);
              array_splice(&$el, 5);

              $cognome = pg_escape_string($el['cognome']);
              $query = "UPDATE persone SET cognome = '{$cognome}'  ".
                "WHERE id_persona = '$id_p' AND id_si = '$id' ".
                    "AND cognome <> '{$cognome}' ;";
              executeQuery($db, $query);

              $nome = pg_escape_string($el['nome']);
              $query = "UPDATE persone SET nome = '{$nome}' WHERE id_persona = $id_p' ".
              "AND id_si = '$id' AND nome <> '{$nome}' ;";
              executeQuery($db, $query);

              $snome = pg_escape_string($el['secondoNome']);
              $query = "UPDATE persone SET secondo_nome = '{$snome}'  ".
                "WHERE id_persona = '$id_p' AND id_si = '$id' ".
                    "AND secondo_nome <> '{$snome}' ;";
              executeQuery($db, $query);
```

```
                $query = "UPDATE persone SET chiave = '{$el['codiceFiscale']}' ".
                  "WHERE id_persona = '$id_p' AND id_si = '$id' ".
                      "AND chiave <> '{$el['codiceFiscale']}' ;";
                executeQuery($db, $query);
              }
 // Check dati da single)person web-service

                $wsdl_url = "http://pimpa.isti.cnr.it/PERSONALE/web-services/people/people.wsdl";
                ini_set("soap.wsdl_cache_enabled", "0");
                $client = new SoapClient($wsdl_url);
                $persona = $client->datiPersona($anag[0]['chiave']);
                // dati singola persona da PERSONALE
                $p = object_to_array_recusive($persona);  //preint_r($p);

                $quit += updateAffiliazioni($p,$id_p);
 //             $quit += checkWebtitle($p,$id_p);
                $quit += checkEmail($p,$id_p);
                $quit += checkStanze($p,$id_p);
                $quit += checkTelefoni($p,$id_p);
            }
        }
        $k++;
 //  if($k == 100) return ;
    }
    echo("<br /> \n Checked $k items <br /> \n");
    return $quit;
 }
 // END function check_Persons

 /* Update affiliazioni */
 function updateAffiliazioni($p,$id_p){
    global $db, $ucodes;

    $quit = 0;
   // Consiglio Istituto codice 29 da fare a mano sul db del sito
    $query = "SELECT at.id_attivita, at.id_unit, u.codice ".
        "FROM attivita at, units u, area_units au, aree a ".
        "WHERE u.id_unit = at.id_unit AND au.id_unit = u.id_unit AND au.id_area = a.id_area ".
 //      "AND ( a.tipo = 'RAD' OR a.tipo = 'ISER' ) ".
       "AND au.fine IS NULL  AND u.fine IS NULL AND u.codice != '29' ".
        "AND at.id_persona = '$id_p' AND at.tipo = 'affiliazione' AND at.fine IS NULL ;";
    $old = getRecords($db,$query);
    $aff = $p["gruppi"];  // preint_r($p); preint_r($aff);

    if(isset($aff) && !empty($aff)){
        remapUnits(&$aff);

        if(isset($old) && !empty($old)) {      // active old affiliations
            $ocod = arrayColumn($old, 'codice');
            $ocanc = array_diff($ocod,$aff);
            $oadd = array_diff($aff, $ocod); // var_dump($aff, $aff1, $ocod, $ocanc, $oadd);
            // echo("<br> \n");
        } else
        {      // no old active affiliation
            $oadd = $aff;  // preint_r($oadd);
            $ocanc = array_diff($aff,$aff);  // no cancel
        }

        foreach($ocanc as $d){
            $idu = $ucodes[$d]['id_unit'];
            if(count($idu) != 1){
                preint_r($d); preint_r($idu);
                echo("{$p['nome']} {$p['cognome']} $id_p cancel affiliation: $d ".
                " (id_unit=$idu ) is an unknown or ambiguous unit  <br /> \n");
            } else
            {
                if($idu != 31){     // evita di cancellare il Direttore (il sito usa una unit DIR )
                    $query ="UPDATE attivita SET fine = CURRENT_DATE WHERE id_persona = '$id_p'  ".
                                "AND id_unit = '$idu' ;";  // preint_r($query);
                    executeQuery($db, $query);
                    echo("{$p['nome']} {$p['cognome']}} dropped from $d->$idu <br /> \n");
                }
            }
        }

        foreach($oadd as $d){
            $idu = $ucodes[$d]['id_unit'];  //var_dump($oadd, $idu);
            if(count($idu) != 1){
                echo("{$p['nome']} {$p['cognome']} $id_p add affiliation: $d (id_unit=$idu ) ".
                "is an unknown or ambiguous unit <br /> \n");
            } else
            {
                $query = "INSERT into attivita (id_persona, tipo, inizio, mesi_anno, id_unit) ".
                            "VALUES ('$id_p', 'affiliazione', CURRENT_DATE, '12', '$idu' ) ;";
                  // preint_r($query);
                executeQuery($db, $query);
                echo("{$p['nome']} {$p['cognome']} added to $d->$idu <br /> \n");
            }
        }
```

```
  } else
  {
     echo("NO AFFILIATION for {$p['nome']} {$p['cognome']} $id_p <br /> \n");
  }

  if(0 < (count($oadd)+count($ocanc))) {
  echo("TO CECK activity time distribution for {$p['nome']} {$p['cognome']}  ".
  "id_persona = $id_p  <br /> \n");
  }

}
//  END function updateAffiliazioni

/* Check webtitles */
function checkWebtitle($p,$id_p){
  global $db;

  $quit = 0;
  $id = $p['id'];

  $query = "SELECT nome, cognome, webtitle FROM persone  ".
                "WHERE id_persona = '$id_p' AND id_si = '$id' ;";
  $anag = getRecords($db,$query);

  $common = similar_text($anag[0]['webtitle'], $p['english'], $percent);
  if($percent < 70.0) {
  echo("CECK dbweb webtitle (old {$anag[0]['webtitle']} vz new {$p['english']}) of ".
  "{$p['nome']} {$p['cognome']}  id_persona = $id_p <br /> \n");
  }
  /*
  $query = "UPDATE persone SET webtitle = '{$p['english']}' ".
    "WHERE id_persona = '$id_p' AND id_si = '$id' ".
       "AND (webtitle <> '{$p['english']}' OR webtitle IS NULL) ;";

  echo $query."<br/>\n";

  executeQuery($db, $query);
  */
  if($p['english'] == 'ISTI Friend'){
     $query = "UPDATE persone SET interno = FALSE  WHERE id_persona = '$id_p' ".
       "AND id_si = '$id' ;";
     executeQuery($db, $query);
  }
  return $quit;
}

/* Check emails */
function checkEmail($p,$id_p){
  global $db;

  $quit = 0;
  $id = $p['id'];
  $temail = '/^[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*(\.[a-z]{2,3})$/';
  $remail = trim($p['email']);
  if(!preg_match($temail, $remail))
  {
    echo("Invalid or Missing EMAIL ( $remail ) for {$p['nome']} {$p['cognome']}  ".
    "id_persona = $id_p  <br /> \n");
  }
  else
  {
    $query = "SELECT email FROM persone WHERE id_persona = '$id_p' AND id_si = '$id' ;";
    $anag = getRecords($db,$query);

    $common = similar_text($anag[0]['email'], $p['email'], $percent);
    if($percent < 95.0 && $percent > 85.0)
    {
        echo("CECK email ({$anag[0]['email']}) of {$p['nome']} {$p['cognome']}  ".
        "id_persona = $id_p $percent <br /> \n");
    }
    elseif($percent < 85.0)
    {
      $update = "UPDATE persone SET email = '$remail' WHERE id_persona = '$id_p' ".
        "AND id_si = '$id' ;";
      executeQuery($db, $update);
      echo("EMAIL '$remail' to  {$p['nome']} {$p['cognome']}  id_persona = $id_p  ".
      "<br /> \n");
    }
  }
  return $quit;
}

/* Check stanze */
function checkStanze($p,$id_p)
{
  global $db;

  $quit = 0;
  $id = $p['id']; //preint_r($p);
```

```
   $stanze = $p['stanze'];  //preint_r($stanze);
   if(empty($stanze) || !isset($stanze))
   {
     echo("Missing STANZE for {$p['nome']} {$p['cognome']}  id_persona_si = $id ".
     " <br /> \n");
     $quit++;
   }
   else
   {
     $query = "SELECT id_stanza FROM logistica WHERE id_persona = '$id_p' ".
        "AND fine IS NULL ;";
     $olg = getRecords($db,$query); // preint_r($olg);
     $olg = arrayColumn($olg, 'id_stanza');

     $troom = '/^[A|C|I]\-[0-9]+[a-z]*$/';
     foreach($stanze as $s)  // stanze importate
     {
       $st = trim($s['stanza']);  //preint_r($st);
       switch($st)
       {
         case 'Biblio':
           $nlg = array('id_stanza'=>'Library');
         break;
         case 'Area':
           $nlg = array('id_stanza'=>'Area');
         break;
         default:
           if(preg_match($troom, $st))
             {
               $nlg = array('id_stanza'=>$st);
             }
             else
             {
                 echo("Invalid STANZA ( $st ) for  {$p['nome']} {$p['cognome']}  ".
                 "id_persona = $id_p  <br /> \n");
             }
       }
     }

//      preint_r($nlg);
//      preint_r($olg);
     if(isset($nlg) && !empty($nlg)) // assegnate stanze
     {
       if(isset($olg) && !empty($olg)) { //active stanze
       $ocanc = array_diff($olg,$nlg);
       $oadd = array_diff($nlg, $olg);
       }
       else
       {     // no old active affiliation
         $oadd = $nlg;  // preint_r($oadd);
         $ocanc = array_diff($nlg,$nlg);  // no cancel
       }

       foreach($ocanc as $i=>$dn)
       {
         $query = "UPDATE logistica SET fine = CURRENT_DATE WHERE id_persona = '$id_p' ".
             "AND id_stanza = '$dn' ;";
         executeQuery($db, $query);
         echo("Deleted STANZA '$dn' for  {$p['nome']} {$p['cognome']}  ".
         "id_persona = $id_p  <br /> \n");
       }

       foreach($oadd as $k=>$in)
       {
         $query = "SELECT id_stanza FROM stanze WHERE id_stanza = '$in' ;";
         $ist = getRecords($db, $query);
         if(empty($ist) || !isset($ist))
         {
           echo("NO STANZA ( $in ) in db web for {$p['nome']} {$p['cognome']} ".
           " id_persona_si = $id  <br /> \n");
           $quit++;
         }
         else
         {
           $query = "INSERT INTO logistica (id_persona, id_stanza, inizio, fine) ".
             "VALUES ('$id_p', '$in', CURRENT_DATE, NULL );";
           executeQuery($db, $query);
           echo("STANZA '$in' for  {$p['nome']} {$p['cognome']}  id_persona = $id_p  ".
           "<br /> \n");
         }
       }

     }
   }
   return $quit;
 }
 // END check stanze

 /* Check telefoni */
```

```
function checkTelefoni($p,$id_p)
{
  global $db;

  $quit = 0;
  // Update telefoni
  $id = $p['id']; // id_si della persona
  $tf = array();
  $tm = array();
//  preint_r($p);
  $telefoni = $p['recapitiTelefonici'];  // preint_r($telefoni);
  if(empty($telefoni) || !isset($telefoni))
  {
    echo("Invalid or missing TELEFONI for {$p['nome']} {$p['cognome']}  ".
    "id_persona_si = $id  <br /> \n");
    $quit++;
  }
  else
  {
    // telefoni interni e mobiles assegnati nel db del sito
    $query = "SELECT numero FROM telefoni_persone WHERE id_persona = '$id_p' ".
        "AND tipo = 'interno' AND fine IS NULL; ";
    $otf = getRecords($db, $query);
    $otf = arrayColumn($otf, 'numero');
    $query = "SELECT numero FROM telefoni_persone WHERE id_persona = '$id_p' ".
        "AND tipo = 'mobile' AND fine IS NULL; ";
    $otm = getRecords($db, $query);
    $otm = arrayColumn($otm, 'numero');

    $pinterno = '/^[238][0-9]{3}$/';
//    $interno_full = '/^\+39[\ \-]050[\ \-]315[0-9]{4}$/';
//    $aziendale = '/^348[\ \-]*[0-9]+$/';
    $pmobile = '/^3[0-9]{2}[\ ]*[0-9]+$/';

    foreach($telefoni as $t)  // telefoni importati
    {
      switch($t['tipoRecapitoTelefonico'])
      {

      case 'Ufficio':
        $tu = trim($t['numero']);
        $sep = strlen($tu)-4;
        $tu = substr($tu, $sep);
        if(!preg_match($pinterno, $tu)) {
          echo("Invalid TELEFONO INTERNO '$tu' for {$p['nome']} {$p['cognome']}  ".
          "id_persona_si = $id  <br /> \n");
          continue;
        } else {
          $tf = array('numero'=>$tu);
        }
      break;

      case 'Mobile aziendale':
        $tu = substr(trim($t['numero']), 4);
        if(!preg_match($pmobile, $tu)) {
          echo("Invalid MOBILE AZIENDALE '$tu' for {$p['nome']} {$p['cognome']}  ".
          "id_persona_si = $id  <br /> \n");
          continue;
        } else {
          $tm = array('numero'=>$tu);
        }
      break;

      default:
        echo("Missing or invalid {$t['tipoRecapitoTelefonico']} for {$p['nome']} ".
        "{$p['cognome']} id_persona_si = $id  <br /> \n");
      }
    }
//    preint_r($tf); preint_r($tm);

  // cancellazioni e riassegnazioni
    if(isset($tf) && !empty($tf)) //  ci sono variazioni dei telefoni interni
    {
      if(isset($otf) && !empty($otf)) // telefoni attivi
      {
        $ocanc = array_diff($otf,$tf);
        $oadd = array_diff($tf, $otf);
      }
      else
      {     // no telefoni attivi
          $oadd = $tf;  // preint_r($oadd);
          $ocanc = array_diff($tf,$ntf);  // no cancel
      }
      // cancellazioni
      foreach($ocanc as $i=>$dn) {
        $query = "UPDATE telefoni_persone SET fine = CURRENT_DATE  WHERE id_persona = '$id_p' ".
                 "AND tipo = 'interno' AND numero = '$dn' ;";
        executeQuery($db, $query);
        echo("Deleted TEL INTERNO ( $dn ) for {$p['nome']} {$p['cognome']}  ".
        "id_persona = $id_p  <br /> \n");
```

```
      }
      // aggiunte
      foreach($oadd as $k=>$in)
      {
        $query = "INSERT INTO telefoni_persone (id_persona, numero, tipo, inizio) ".
                 "VALUES ('$id_p', '$in', 'interno', CURRENT_DATE);";
        executeQuery($db, $query);
        echo("TEL INTERNO ( $in ) for  {$p['nome']} {$p['cognome']}  id_persona = $id_p ".
        " <br /> \n");
      }
    }  // chiude aggiornamento telefoni

    if(isset($tm) && !empty($tm)) { // ci sono variazioni nei mobile
      if(isset($otm) && !empty($otm))
      { // mobile attivi
          $ocanc = array_diff($otm,$tm);
          $oadd = array_diff($tm, $otm);
      } else
      {  // no mobile attivi
          $oadd = $tm;  // preint_r($oadd);
          $ocanc = array_diff($tm,$ntm);  // no cancel
      }
      // cancellazioni
      foreach($ocanc as $i=>$dn) {
        $query = "UPDATE telefoni_persone SET fine = CURRENT_DATE  WHERE id_persona = '$id_p' ".
                 "AND tipo = 'mobile' AND numero = '$dn' ;";
        executeQuery($db, $query);
        echo("Deleted MOBILE ( $dn ) for {$p['nome']} {$p['cognome']}  id_persona = $id_p  ".
        "<br /> \n");
      }
      // aggiunte
      foreach($oadd as $k=>$in) {
        $query = "INSERT INTO telefoni_persone (id_persona, numero, tipo, inizio) ".
                 "VALUES ('$id_p', '$in', 'mobile', CURRENT_DATE);";
        executeQuery($db, $query);
        echo("MOBILE ( $in ) for  {$p['nome']} {$p['cognome']}  id_persona = $id_p  ".
        "<br /> \n");
      }
    } //chiude aggiornamento mobile
  }  //chiude aggionamenti
  return $quit;
}
//  END checkTelefoni

// Check to asssign webtitle manually
function checkWebtitlesAll(){
  global $db;
    echo(" </p> \n <h4>Persone a cui assegnare webtitle</h4> \n <p> \n" );
    $query = "SELECT p.cognome, p.nome FROM persone p WHERE p.webtitle IS NULL ".
        "AND p.interno IS TRUE ;";
    $nowebtitle = getRecords($db, $query);
    if($nowebtitle) {
        foreach($nowebtitle as $e) {
            echo("{$e['cognome']}  {$e['nome']} <br /> \n");
        }
    }
}

// further check on interno = true: a person without an activity !! possible errors
function checkAttiviAll() {
  global $db;
    echo(" </p> \n <h4>Persone che non compaiono in attivita (affiliaz. etc.) - ".
    "Errore ?</h4> \n <p> \n" );
    $query = "SELECT p.id_persona, p.cognome, p.nome FROM persone p ".
      "WHERE p.interno IS TRUE ".
          "AND NOT EXISTS (SELECT a.id_persona FROM attivita a ".
          "WHERE p.id_persona = a.id_persona AND a.fine IS NULL ) ;";
    $noactive = getRecords($db, $query);
    if($noactive) {
        foreach($noactive as $e) {
            echo("{$e['cognome']}  {$e['nome']} <br /> \n");
//          $id = $e['id_persona'];
//          $query = "UPDATE persone SET interno = FALSE WHERE id_persona = $id";
//          executeQuery($db, $query);
        }
    }
}

/* Insert  da web-services id_si into persone sul db del sito
(per check  o maintenance) */
function IdSiPersons(){
  global $db;

  $wsdl_url = "http://pimpa.isti.cnr.it/PERSONALE/web-services/people/people.wsdl";
  ini_set("soap.wsdl_cache_enabled", "0");
  $client = new SoapClient($wsdl_url);


  /* nome, secondoNome, cognome, codiceFiscale, profilo, subprofilo,
  english, gruppi (array), id  */
```

```
  $elenco = $client->personale();
  $elenco = object_to_array_recusive($elenco);
  foreach($elenco as $e) {
      $cf = $e['codiceFiscale'];
      $id = $e['id'];
      if(!checkCodFisc($cf)){
          echo("Codice fiscale origine non valido:   <br /> \n");
      } else
      {
          $query = "SELECT  id_persona, cognome, nome  FROM persone WHERE chiave = '$cf' ;" ;
          $persone = getRecords($db, $query);
          if(count($persone) !== 1) {
              echo("Persone con lo stesso codice fiscale su db_sito, o persona non trovata: ".
              "{$e['codiceFiscale']} {$e['cognome']} {$e['nome']} ");
              preint_r($persone);
          } else
          {
              $query = "UPDATE persone SET id_si = '$id' WHERE  chiave = '$cf' ;";
//            echo("$query <br /> \n");
              $ret = executeQuery($db, $query);
              echo("$cf - $id  <br /> \n");
          }
      }
  }
}
// END IdSiPersone

?>
```

Funzioni frequentemente usate.

```
<?php
// mb_internal_encoding( 'UTF-8' );

$db = pg_connect('host=<host> dbname=<dbname> user=<userid> password=<password>') or die();

function getRecords($d, $query) {
  return postgresGetRecords($d, $query);
}

function prepare($db, $name, $query) {
  return pg_prepare($db, $name, $query);
}

function execute($db, $name, $values) {
  return pg_execute($db, $name, $values);
}

function executeQuery($d, $query) {
  return postgresExecuteQuery($d, $query);
}

function updateOnDuplicate($d,  $table, $key, $variables) {
  postgresUpdateOnDuplicate($d,  $table, $key, $variables);
}

function mapRecords($db, $query, $var = NULL) {
  if(!$var) $var = 'id';
  $result = pg_query($db, $query);
  if (!$result) {
    echo "Problem with query " . $query . "<br/>";
    echo pg_last_error();
    exit();
  }

  $ret = Array();
  while($row = pg_fetch_assoc($result)) {
    $ret[$row[$var]] = $row;
  }
  return $ret;
}


function postgresGetRecords($db, $query) {
  $result = pg_query($query);
  if (!$result) {
    echo "Problem with query " . $query . "<br/>";
    echo pg_last_error();
    exit();
  }
  $ret = Array();
  while($row = pg_fetch_assoc($result)) {
    $ret[] = $row;
  }
  return $ret;
}
```

```
function postgresExecuteQuery($db, $query) {
  $res =  pg_query($query);
  if(!$res) die ('Query error: <br/>'.$query);
  return $res;
}

function postgresGetRecord($ignoreme, $query) {
  $result = pg_query($query);
  if (!$result) {
    echo "Problem with query " . $query . "<br/>";
    echo pg_last_error();
    exit();
  }
  return pg_fetch_assoc($result);
}

function postgresUpdateOnDuplicate($db, $table, $key, $variables) {
  $update = "UPDATE $table SET ";
  $eq = array();
  $fields = array();
  $values = array();
  foreach($variables as $field=>$value) {
    if($field != $key)
      $eq[] = "$field = $value";
    $fields[] = $field;
    $values[] = "$value as $field";
  }
  $update .= join(',', $eq)." WHERE $key={$variables[$key]};";
  postgresExecuteQuery($db, $update);
  $insert = "INSERT INTO $table (".join(',', $fields).") (SELECT "
    .join(',', $values)." WHERE NOT EXISTS ".
    "(SELECT 1 FROM $table WHERE $key={$variables[$key]}));";
  postgresExecuteQuery($db, $insert);
}

// Check italian Codice Fiscale
function checkCodFisc($cod)
{
  $pattern = '/^[A-Z]{6}[\d]{2}[A-Z][\d]{2}[A-Z][\d]{3}[A-Z]$/';
  if((strlen($cod) != 16) || (!preg_match($pattern, $cod)))  {
       return false;
  } else {
      return true;
  }
}

function preint_r($array)
   {
      echo '<pre>';
      print_r($array);
      echo '</pre>';
   }

/* return id_persona registrata from id_si  and check unique */
function idsi_to_idpers_uniq($id_si)
{       global $db;

    $query = "SELECT id_persona FROM persone WHERE id_si = $id_si ;";
    $idp = getRecords($db,$query);
    if(count($idp) != 1){
    $idp = 0;
    } else {
    $idp = intval($idp[0]['id_persona']);
    }
    return $idp;
}

/* return id_persona interna from id_si  and check unique */
function idsi_to_idpers_int_uniq($id_si)
{       global $db;

    $query = "SELECT id_persona FROM persone WHERE id_si = $id_si AND interno = TRUE ;";
    $idp = getRecords($db,$query);
    if(count($idp) != 1){
    $idp = 0;
    } else {
    $idp = intval($idp[0]['id_persona']);
    }
    return $idp;
}

/* return id_web from id_si  and check unique */
function idsi_to_idweb_uniq($id_si,$table,$column_key)
{       global $db;

    $query = "SELECT '$column_key' FROM '$table' WHERE id_si = '$id_si' ;";
    $idt = getRecords($db,$query);
    if(count($idt) != 1){
    $idt = 0;
    } else {
```

```
    $idt = intval($idt[0][$column_key]);
    }
    return $idt;
}

/* Utilities per id_areas vz codici */
function getAreas() {
  global $db;
  $query = "SELECT id_area, codice FROM aree ".
           "WHERE fine IS NULL ";
  return mapRecords($db, $query, 'codice');
}

/* Utilities per id_units vz codici */
function getUnits() {
  global $db;
  $query = "SELECT u.id_unit, u.codice FROM units u, area_units au, aree a ".
           "WHERE au.id_unit = u.id_unit AND au.id_area = a.id_area ".
//         "AND ( a.tipo = 'RAD' OR a.tipo = 'ISER' ) ".
           "AND au.fine IS NULL  AND u.fine IS NULL AND a.fine IS NULL ";
  return mapRecords($db, $query, 'codice');
}

function remapUnits(&$gruppi) {
  foreach($gruppi as &$u) {
    switch($u) {
          case 'CI': unset($u); break;
      case 'SAS - PC': $u = 'SAS'; break;
      case 'SAS - SS': $u = 'SAS'; break;
      case 'SSIF - SI': $u = 'SSIF'; break;
      case 'GL - SI': $u = 'Sysinf'; break;
      case 'SSIF - FS': $u = 'SSIF'; break;
    }
  }
  unset($u);
}

/*Simulate array_column of php >= 5.5  (from Internet) */
  function arrayColumn(array $array, $column_key, $index_key=null)
  {
        if(function_exists('array_column ')){
            return array_column($array, $column_key, $index_key);
        }
        $result = array();
        foreach($array as $arr){
            if(!is_array($arr)) continue;
            if(is_null($column_key)){
                $value = $arr;
            }else{
                $value = $arr[$column_key];
            }
            if(!is_null($index_key)){
                $key = $arr[$index_key];
                $result[$key] = $value;
            }else{
                $result[] = $value;
            }
        }
        return $result;
  }

/*  Change an object as $key => $val into an array of arrays  (from Internet) */
  function object_to_array_recusive ( $object, $assoc=TRUE, $empty='' )
  {
    $res_arr = array();
    if (!empty($object)) {
        $arrObj = is_object($object) ? get_object_vars($object) : $object;
        $i=0;
        foreach ($arrObj as $key => $val) {
            $akey = ($assoc !== FALSE) ? $key : $i;
            if (is_array($val) || is_object($val)) {
                $res_arr[$akey] = (empty($val)) ? $empty : object_to_array_recusive($val);
            }
            else {
                $res_arr[$akey] = (empty($val)) ? $empty : (string)$val;
            }
        $i++;
        }
    }
    return $res_arr;
  }

function delete_col(&$array, $offset) {
    return array_walk($array, function (&$v) use ($offset) {
        array_splice($v, $offset, 1);
    });
}

function change_key($array, $old_key, $new_key) {
```

```
    if( ! array_key_exists( $old_key, $array ) )
        return $array;

    $keys = array_keys( $array );
    $keys[ array_search( $old_key, $keys ) ] = $new_key;

    return array_combine( $keys, $array );
}

/*  a function similar to pg_insert in PHP (only output sql statement) - from Internet */
function db_mount_insert($table,$array)
{
    $str = "insert into $table (";
    while(list($name,$value) = each($array)) {
        $str .= "$name,";
    }
    $str[strlen($str)-1] = ')';
    $str .= " values (";
    reset($array);
    while(list($name,$value) = each($array)) {
        if(is_string($value))
            $str .= "'$value',";
        else
            $str .= "$value,";
    }
    $str[strlen($str)-1] = ')';
    $str .= ";"    ;
    return $str;
   }

?>
```

Un esempio del suo uso.

Il comando usato per avviare il programma descritto e tipicamente:

```
wget
```

Un esempio di output dall'aggiornamento è il seguente:

```
Dbname: isti{utf8 - Importing PERSONALE data from webservices on 2016-01-25 05:20:03
(i dati del Direttore vanno inseriti e corretti direttamente sul db)

Counts Step_1 - elenco: 299 299 attivi: 297 297 registrati: 364 364 canc: 0 add: 0 react: 2 check: 297
Reactivated Massimo Mannocci id_si = 218
Reactivated Riccardo Medves id_si = 512
Counts - elenco: 299 299 attivi: 299 299 registrati: 364 364 canc: 0 add: 0 react: 0 check: 299

Counts Step_2 - elenco: 299 299 attivi: 299 299 registrati: 364 364 canc: 0 add: 0 react: 0 check: 299
Invalid or missing TELEFONI for Paola Andriani id_persona_si = 450
Invalid or Missing EMAIL ( ) for Viola Bachini id_persona = 717
Invalid or missing TELEFONI for Viola Bachini id_persona_si = 532
Missing STANZE for Laura Baldini id_persona_si = 543
Invalid or missing TELEFONI for Laura Baldini id_persona_si = 543
Invalid or Missing EMAIL ( ) for Alessandro Baris id_persona = 707
Invalid or missing TELEFONI for Alessandro Baris id_persona_si = 525
Missing STANZE for Davide Basile id_persona_si = 516
Invalid or missing TELEFONI for Davide Basile id_persona_si = 516
Renzo Beltrame 28 add affiliation: Sysinf (id_unit= ) is an unknown or ambiguous unit
TO CECK activity time distribution for Renzo Beltrame id_persona = 28
Missing STANZE for Filippo Benedetti id_persona_si = 517
Invalid or missing TELEFONI for Filippo Benedetti id_persona_si = 517
Invalid or missing TELEFONI for Giovanni Bonocore id_persona_si = 559
Invalid or missing TELEFONI for Luca Bortolussi id_persona_si = 35
Missing STANZE for Fabio Carrara id_persona_si = 534
Invalid or missing TELEFONI for Fabio Carrara id_persona_si = 534
Invalid or missing TELEFONI for Matteo Catena id_persona_si = 494
Invalid or missing TELEFONI for Mauro Coletto id_persona_si = 463
Missing STANZE for Livia Couto Rubak Rodriguez id_persona_si = 497
Invalid or missing TELEFONI for Livia Couto Rubak Rodriguez id_persona_si = 497
Missing STANZE for Said Daoudagh id_persona_si = 95
Roberta Diciotti 110 add affiliation: Sysinf (id_unit= ) is an unknown or ambiguous unit
TO CECK activity time distribution for Roberta Diciotti id_persona = 110
Invalid or missing TELEFONI for Michela Donati id_persona_si = 560
Invalid or Missing EMAIL ( ) for Daniele Fadda id_persona = 738
Invalid or missing TELEFONI for Daniele Fadda id_persona_si = 554
Invalid or missing TELEFONI for Fabio Falcini id_persona_si = 500
NO AFFILIATION for Luca Frosini 482
Invalid TELEFONO INTERNO '9286' for Daniela Giorgi id_persona_si = 162
Missing STANZE for Valerio Grossi id_persona_si = 167
```

```
Invalid or missing TELEFONI for Valerio Grossi id_persona_si = 167
Invalid or missing TELEFONI for Riccardo Guidotti id_persona_si = 410
Invalid or missing TELEFONI for Hanna Kavalionak id_persona_si = 476
Invalid or Missing EMAIL ( ) for Yuri Lacerda id_persona = 748
Invalid or missing TELEFONI for Yuri Lacerda id_persona_si = 562
Invalid or missing TELEFONI for Giuseppe Lipari id_persona_si = 449
Invalid or missing TELEFONI for Baobao Liu id_persona_si = 551
Invalid or missing TELEFONI for Alessandro Lulli id_persona_si = 462
NO AFFILIATION for Massimo Mannocci 418
Missing STANZE for Giulio Masetti id_persona_si = 549
Invalid or missing TELEFONI for Giulio Masetti id_persona_si = 549
Invalid or Missing EMAIL ( ) for Fabrizio Matarese id_persona = 746
Missing STANZE for Fabrizio Matarese id_persona_si = 561
Invalid or missing TELEFONI for Fabrizio Matarese id_persona_si = 561
Invalid or Missing EMAIL ( ) for Riccardo Medves id_persona = 512
Missing STANZE for Riccardo Medves id_persona_si = 512
Invalid or missing TELEFONI for Riccardo Medves id_persona_si = 512
Invalid or missing TELEFONI for Daniele Metilli id_persona_si = 529
Invalid or Missing EMAIL ( ) for Ioanna Miliou id_persona = 718
Invalid or missing TELEFONI for Ioanna Miliou id_persona_si = 533
TO CECK activity time distribution for Claudio Montani id_persona = 211
Invalid or missing TELEFONI for Vinicius Monteiro De Lira id_persona_si = 428
Daniela Mulas 214 add affiliation: Sysinf (id_unit= ) is an unknown or ambiguous unit
TO CECK activity time distribution for Daniela Mulas id_persona = 214
Invalid or Missing EMAIL ( ) for Umberto Napolitano id_persona = 735
Invalid or missing TELEFONI for Umberto Napolitano id_persona_si = 550
Pasquale Pagano 226 add affiliation: Sysinf (id_unit= ) is an unknown or ambiguous unit
TO CECK activity time distribution for Pasquale Pagano id_persona = 226
Missing STANZE for Luca Pappalardo id_persona_si = 278
Missing STANZE for Luca Pardini id_persona_si = 522
Invalid TELEFONO INTERNO '0419' for Luca Pardini id_persona_si = 522
Invalid or missing TELEFONI for Parvaneh Parvin id_persona_si = 531
Invalid or Missing EMAIL ( ) for Roberto Pavone id_persona = 739
Invalid or missing TELEFONI for Roberto Pavone id_persona_si = 553
Invalid or missing TELEFONI for Costantino Perciante id_persona_si = 555
Invalid or Missing EMAIL ( ) for Guglielmo Piacentini id_persona = 721
Invalid or missing TELEFONI for Guglielmo Piacentini id_persona_si = 535
Invalid or missing TELEFONI for Marco Potenziani id_persona_si = 389
Jonathan Puccini 415 add affiliation: Sysinf (id_unit= ) is an unknown or ambiguous unit
TO CECK activity time distribution for Jonathan Puccini id_persona = 415
Invalid or Missing EMAIL ( ) for Igo Ramalho Brilhante id_persona = 520
Invalid or missing TELEFONI for Igo Ramalho Brilhante id_persona_si = 313
Invalid or Missing EMAIL ( ) for Laura Ricci id_persona = 276
Missing STANZE for Laura Ricci id_persona_si = 304
Invalid or missing TELEFONI for Laura Ricci id_persona_si = 304
Invalid or Missing EMAIL ( ) for Eustachio Rinaldelli id_persona = 683
Invalid or missing TELEFONI for Eustachio Rinaldelli id_persona_si = 506
Invalid or Missing EMAIL ( ) for Salvatore Ruggieri id_persona = 286
Invalid or Missing EMAIL ( ) for Manuele Sabbadin id_persona = 706
Invalid or missing TELEFONI for Manuele Sabbadin id_persona_si = 524
CECK email (antonio.schiavone@isti.cnr.it) of Antonio Schiavone id_persona = 468 86.567164179104
Invalid or Missing EMAIL ( ) for Filippo Simini id_persona = 513
Missing STANZE for Filippo Simini id_persona_si = 513
Invalid or missing TELEFONI for Filippo Simini id_persona_si = 513
Invalid or missing TELEFONI for Eliana Siotto id_persona_si = 351
Missing STANZE for Venkatapathy Subramanian id_persona_si = 507
Invalid or missing TELEFONI for Venkatapathy Subramanian id_persona_si = 507
Invalid or Missing EMAIL ( ) for Ilaria Tono id_persona = 711
Invalid or missing TELEFONI for Ilaria Tono id_persona_si = 527
Invalid or Missing EMAIL ( ) for Luca Ussi id_persona = 743
Invalid or missing TELEFONI for Luca Ussi id_persona_si = 556
Invalid or Missing EMAIL ( Farzad.Vaziri@isti.cnr.it ) for Farzad Vaziri id_persona = 729
Invalid or missing TELEFONI for Farzad Vaziri id_persona_si = 544
Invalid or missing TELEFONI for Maria Venianaki id_persona_si = 521
Invalid or Missing EMAIL ( ) for Paolo Verdini id_persona = 733
Invalid or missing TELEFONI for Paolo Verdini id_persona_si = 545
Federico Volpini 417 add affiliation: Sysinf (id_unit= ) is an unknown or ambiguous unit
TO CECK activity time distribution for Federico Volpini id_persona = 417
Missing STANZE for Jun Xiao id_persona_si = 509
Franco Zoppi 347 add affiliation: Sysinf (id_unit= ) is an unknown or ambiguous unit
TO CECK activity time distribution for Franco Zoppi id_persona = 347

Checked 299 items
Counts - elenco: 299 299 attivi: 297 297 registrati: 364 364 canc: 0 add: 0 react: 2 check: 297

Persone che non compaiono in attivita (affiliaz. etc.) - Errore ?

Persone a cui assegnare webtitle
```

L'output contiene sia informazioni sulle operazioni fatte sia un promemoria per operazioni da fare.