



LUND UNIVERSITY

Ledfunk-databasen

Bonnier, Mikael

2005

[Link to publication](#)

Citation for published version (APA):

Bonnier, M. (2005). *Ledfunk-databasen*. (Mikael Bonnier's Place; Vol. 2). Mikael O. Bonnier.
<http://www.df.lth.se/~mikaelb/ledfunk/>

Total number of authors:

1

General rights

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Ledfunk-databasen

Författare: Mikael Bonnier <mikael.bonnier@med.lu.se>

Databasadministratör
Medicinsk mikrobiologi
Lunds universitet
Utförd: Hösten 2004

20 april 2005

1 Installation

I denna avdelning går vi steg för steg igenom installationen av allt som behövs för att köra Ledfunk-databasen på Windows XP (och troligen även på tidigare och senare Windows-versioner).

1.1 Installera databasmotorn Access

Microsoft Office Access 2003 måste vara installerat. Förmodligen fungerar även tidigare och senare versioner av Access.

1.2 Installera webbservern IIS

Webbservern Internet Information Services (IIS) installeras med hjälp av Kontrollpanelen/Lägg till eller ta bort program/Lägg till/ta bort Windows-komponenter. Då krävs Windows XP Pro CD-ROM. Webbgränssnittet fungerar lika bra med Apache eller Xitami som webbserver.

1.3 Installera programspråket Perl

Perl kan hämtas på <http://www.activestate.com/>. Leta upp och ladda ner senaste versionen av ActivePerl – i skrivande stund ActivePerl 5.8.6.811. Man bör i första hand välja MSI-paketet på 12.6 MB. Man dubbelklickar på filen och följer instruktionerna.

1.4 Installera Perl-modulen Mail-Sender

Vill man kunna skicka e-post till databasadministratören från webbgränssnittet så måste man installera Perl-modulen Mail-Sender. Den installeras genom att köra Perl Packet Manager (PPM) som finns i Start-menyn efter att man installerat ActiveState ActivePerl 5.8. I denna använder man kommandot `install Mail-Sender`.

1.5 Lägg in databasen Ledfunk

Databasen och Perl-program för att läsa in Excel-filer m.m. till databasen finns i zip-filen med ledfunk i mappen `Ledfunk`. Denna mapp placeras någonstans på servern, t.ex. i mappen `My Downloads`.

1.6 Skapa en ODBC-databaskoppling

Perl-programmen använder databasmotorn Access via ODBC (Open Database Connectivity). En ODBC-koppling skapas genom att köra Perl-programmet `createdsn.pl`, se avdelning 3.2.2, möjligen fungerar denna koppling endast för den enskilda användaren. Perl-program körs genom att starta Tillbehör/Kommandotolken, gå till rätt mapp och ange kommandot `perl program.pl`. För att få en ODBC-koppling som fungerar med webbservern måste man gå in i Kontrollpanelen/Administrationsverktyg/ODBC Data Source Administrator på fliken System DSN och skapa en ny datakälla som är en kopia av datakällan `ledfunk` på fliken User DSN. Man bör därefter ta bort `ledfunk` på fliken User DSN.

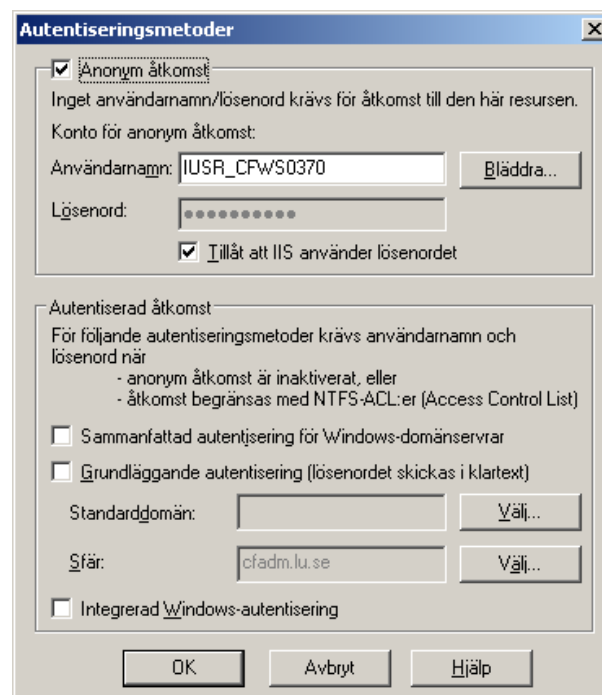
1.7 Konfigurera webbservern IIS

Webbservern IIS måste konfigureras via Kontrollpanelen/Administrationsverktyg/Internet Information Services. Kontrollera att dialogrutan Autentiseringsmetoder liknar figur 1. Man bör testa att man kan komma åt webbservern från andra datorer (men den är spärrad till Lunds universitet). Exempel på adress: `http://130.235.255.255/cgi-bin/login.pl`. Användarnamn och lösenord finns i databasen i tabellen `Anvandare`.

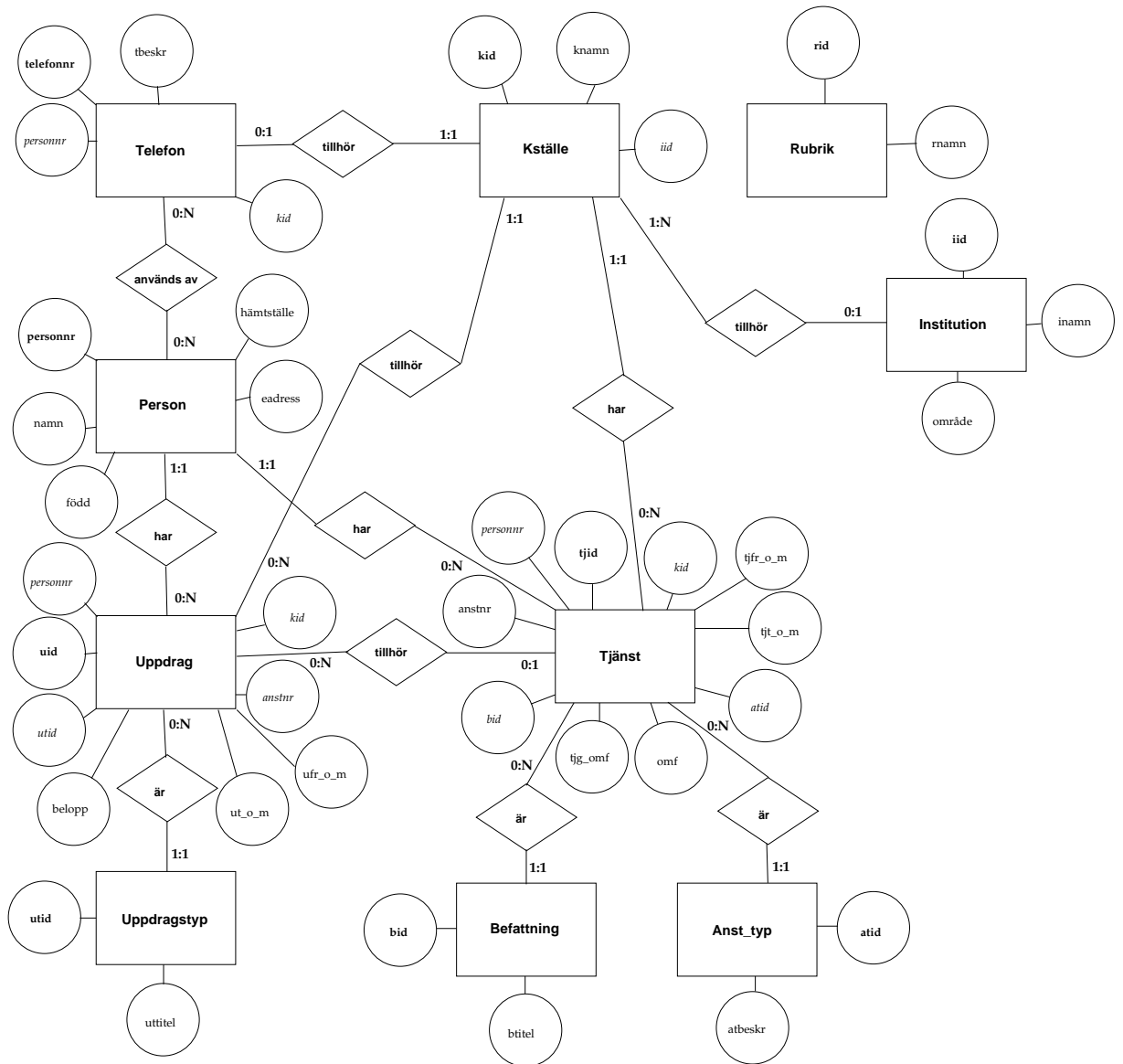
I zip-filen med ledfunk så skall innehållet i mappen `wwwroot` läggas i samma mapp under mappen `Inetpub` på webbservern.

2 Databasdesign

Designen av databasen finns i figur 2.



Figur 1: *Dialogrutan Autentiseringsmetoder.*



Figur 2: ER-diagram över Ledfunk-databasen.

3 Programkod

Denna avdelning är troligen bara av intresse för programmerare, även om en del användbar SQL-kod kan framgå. Man kan även vilja modifiera inställningar i `Check.pm`, se 3.1.2, eller i stilmallen `ledfunk.css`, se 3.1.1.

3.1 Webbgränssnitt

3.1.1 `ledfunk.css`

Här finns stilmallen som kan modifieras för olika behov. Notera att utskrift bör ske på liggande A4.

```
@media all {
body {font-family:verdana, arial, helvetica, sans-serif;
      font-size:16px; margin-left:5px;
      background-color:#C0CEFF; color:#000000;}
tr.long {display:none;}
th, td, select {font-family:verdana, arial, helvetica, sans-serif;
               font-size:13px; margin-left:0px;
               color:#000000;}
p        {font-family:verdana, arial, helvetica, sans-serif;
          font-size:16px; margin-left:0px;
          color:#000000;}
h1       {font-family:verdana, arial, helvetica, sans-serif;
          font-size:50px; font-weight:bold; Margin-left:0px;
          color:#000000;}
a:link   {color:#404040;}
a:visited {color:#909090;}
a:hover  {color:#000000;}
a:active {color:#FFFFFF;}
.small  {font-family:arial, helvetica, sans-serif;
         font-size:10px; margin-left:0px;
         color:#C0CECE;}
}

@media print {
body {font-family:verdana, arial, helvetica, sans-serif;
      font-size:10pt; margin-left:0pt;
      background-color:white; color:black;}
tr.long {display:block; page-break-before:always;}
th, td, select {font-family:verdana, arial, helvetica, sans-serif;
               font-size:6pt; margin-left:0pt;
               color:black;}
h1       {font-family:verdana, arial, helvetica, sans-serif;
          font-size:24pt; font-weight:bold; Margin-left:0pt;
          color:black;}
p        {font-family:verdana, arial, helvetica, sans-serif;
          font-size:12pt; margin-left:0pt;
          color:black;}
input {display:none;}
a:link   {color:#404040; text-decoration:none;}
a:visited {color:#909090; text-decoration:none;}
a:hover  {color:#000000;}
a:active {color:#FFFFFF;}
.small  {display:none;}
}
```

3.1.2 Check.pm

I denna Perl-modul finns funktioner som används av alla andra Perl/CGI-program.

```
#--- Inställningar ---
$Check::from = 'mikael.bonnier@med.lu.se';
$Check::to = 'mikaelb@df.lth.se';
$Check::author = 'mikael.bonnier@med.lu.se';

sub checkRemoteHost {
    my $query = shift @_;

    my $remote_host = $query->remote_host();
    if(!($remote_host =~ /lu\.se$/ || $remote_host =~ /lth\.se$/
        || $remote_host =~ /^130\.235\.\/
        || $remote_host =~ /^194\.47\.250\.\/ || $remote_host =~ /^127.0.0.1$/)) {
        die 'Access from outside lu.se or lth.se forbidden, try using vpn.lu.se.'
            . "remote_host=$remote_host.";
    }
}

#--- Slut på inställningar ---

$Check::db = NULL;

my %headings;
my $query;

sub printHead {
    my $query = shift @_;
    my $title = shift @_;
    my $onload = shift @_;

    print $query->header;

    print $query->start_html(-title=>$title,
                            -author=>$Check::author,
                            -style=>{'src'=>'ledfunk.css'},
                            -onload=>$onload);

    print qq|<h2>$title</h2>|;
}

sub initDb {
    #Define the Data Source Name
    my $DSN = "ledfunk";

    #Create a Win32::ODBC object

    $Check::db = new Win32::ODBC($DSN);

    #Verify the connection is valid
    if(!$Check::db) {
        print "Failed to Connect $DSN\n";
        Win32::ODBC::DumpError();
        #You can't do anything without a connection so die here
        die;
    }
    return $Check::db;
}

sub check {
    my $needed = shift @_;
```

```

$query = new CGI;
checkRemoteHost($query);

initDb();

my $sid;
my $anamn;
my $behorighet = 0;
if($query->param('sid')) {
    $sid = $query->param('sid');

    my $SQLStatement = qq|SELECT sid, anamn, ip, behorighet
        FROM Sessioner, Anvandare
        WHERE Sessioner.aid = Anvandare.aid AND sid = $sid|;
    FireSQL($SQLStatement);
    if($Check::db->FetchRow()) {
        my %hash = $Check::db->DataHash;
        $anamn = $hash{'anamn'};
        $ip = $hash{'ip'};
        if($ip eq $query->remote_addr()) {
            $behorighet = $hash{'behorighet'};
        }
        my ($sec, $min, $hour, $mday, $mon, $year) = localtime();
        my $timenow = ($year+1900) . '-' . ($mon+1) . '-' . $mday . " $hour:$min:$sec";
        $SQLStatement = qq|UPDATE Sessioner
            SET senast = '$timenow'
            WHERE sid = $sid|;
        FireSQL($SQLStatement);
    }
}

if($behorighet < $needed) {
    $Check::db->Close();
    print $query->redirect(-uri=>"login.pl",
        -nph=>0);

    die;
}

FireSQL(qq|SELECT rid, rnamn
    FROM Rubriker|);
while($Check::db->FetchRow()) {
    my %hash = $Check::db->DataHash;
    if(!defined($hash{'rnamn'})) {
        $hash{'rnamn'} = $hash{'rid'};
    }
    $headings{$hash{'rid'}} = $hash{'rnamn'};
}

return ($Check::db, $query, $sid, $anamn);
}

sub FireSQL {
    my $SQL = shift @_;
    if($Check::db->Sql($SQL)) {
        print "Error running statement $SQL";
        Win32::ODBC->DumpError();
    }
}

sub printSQL {
    my $head = 9;

```



```

my $rows = 37;
FireSQL(shift @_);
my $sid = shift @_;
my $rowsExist = $Check::db->FetchRow();
my @fieldNames = $Check::db->FieldNames();
my $fd;
my $i;
print qq|<table border="0" width="100%">|;
print qq|<tr>|;
foreach $fd (@fieldNames) {
    my $heading = $headings{$fd};
    print "<td><b>$heading</b></td>\n"
}
print "</tr>";
my $color = '#CECECE';
for($i = $head; $rowsExist; ++$i, $rowsExist = $Check::db->FetchRow()) {
    if($i % $rows == 0) {
        print qq|<tr class="long">|;
        foreach $fd (@fieldNames) {
            my $heading = $headings{$fd};
            print "<td><b>$heading</b></td>\n"
        }
        print "</tr>";
        my $color = '#CECECE';
    }
    if($color eq '#CECECE') {
        $color = '#COCOC0';
    }
    else {
        $color = '#CECECE';
    }
    print qq|<tr bgcolor="$color">|;
    my %hash = $Check::db->DataHash;
    foreach $fd (@fieldNames) {
        my $value;
        if(defined($hash{$fd})) {
            if($fd eq 'eadress') {
                $value = qq|<a href="mailto:$hash{$fd}">$hash{$fd}</a>|;
            }
            elsif($fd eq 'personnr') {
                $value = qq|<a href="pul.pl\?sid=$sid&txtPersonnr=$hash{$fd}"
                >$hash{$fd}</a>|;
            }
            elsif($fd eq 'kid') {
                $value = qq|<a
                href=
                "chledfunk.pl\?sid=$sid&selInstitution=-1&selKstalle=$hash{$fd}&selUppdragstyp=-1"
                >$hash{$fd}</a>|;
            }
            else {
                $value = $hash{$fd};
            }
        }
        else {
            $value = '&nbsp;';
        }
        print qq|<td>$value</td>|;
    }
    print "</tr>\n";
}
print "</table><br/>";
}

```

```

sub printSQLVt {
    FireSQL(shift @_);
    my $sid = shift @_;
    $Check::db->FetchRow();
    my @fieldNames = $Check::db->FieldNames();
    do {
        print qq|<table border="0" width="100%" id="tblResult">|;
        my $fd;
        my $color = '#CECECE';
        foreach $fd (@fieldNames) {
            my $heading = $headings{$fd};
            my %hash = $Check::db->DataHash;
            my $value;
            if(defined($hash{$fd})) {
                if($fd eq 'eadress') {
                    $value = qq|<a href="mailto:$hash{$fd}">$hash{$fd}</a>|;
                }
                elsif($fd eq 'personnr') {
                    $value = qq|<a href="pul.pl\?sid=$sid&txtPersonnr=$hash{$fd}"
                    >$hash{$fd}</a>|;
                }
                elsif($fd eq 'kid') {
                    $value = qq|<a href=
                    "chledfunk.pl\?sid=$sid&selInstitution=-1&selKstalle=$hash{$fd}&selUppdragstyp=-1"
                    >$hash{$fd}</a>|;
                }
                else {
                    $value = $hash{$fd};
                }
            }
            else {
                $value = '&nbsp;';
            }
            if($color eq '#CECECE') {
                $color = '#COCOCO';
            }
            else {
                $color = '#CECECE';
            }
            print qq|<tr bgcolor="$color"><td width="300"><b>$heading</b></td><td
            >$value</td></tr>\n|;
        }
        print "</table><br/>\n";
    } while($Check::db->FetchRow());
}

sub printSQLCheckbox {
    my $head = 9;
    my $rows = 37;
    FireSQL(shift @_);
    my $sid = shift @_;
    my $rowsExist = $Check::db->FetchRow();
    my @fieldNames = $Check::db->FieldNames();
    unshift @fieldNames, 'checkbox';
    unshift @headings, '';
    my $fd;
    my $i;
    print qq|<table border="0" width="100%">|;
    print '<tr>';
    foreach $fd (@fieldNames) {
        if($fd eq 'utid' || $fd eq 'tjid2' || $fd eq 'uid2') {

```

```

    next;
  }
  my $heading = defined($headings{$fd}) ? $headings{$fd}
    : $fd eq 'checkbox' ? '' : $fd;
  $heading = $heading eq 'checkbox' ? '' : $heading;
  print "<td><b>$heading</b></td>\n"
}
print '</tr>';
my $color = '#CECECE';
my @uidlst;
for($i = $head; $rowsExist; ++$i, $rowsExist = $Check::db->FetchRow()) {
  if($i % $rows == 0) {
    print qq|<tr class="long">;
    foreach $fd (@fieldNames) {
      if($fd eq 'utid' || $fd eq 'tjid2' || $fd eq 'uid2') {
        next;
      }
      my $heading = defined($headings{$fd}) ? $headings{$fd}
        : $fd eq 'checkbox' ? '' : $fd;
      print "<td><b>$heading</b></td>\n"
    }
    print '</tr>';
    my $color = '#CECECE';
  }
  if($color eq '#CECECE') {
    $color = '#COCOCO';
  }
  else {
    $color = '#CECECE';
  }
}
print qq|<tr bgcolor="$color">;
my %hash = $Check::db->DataHash();
foreach $fd (@fieldNames) {
  my $value;
  if($fd eq 'utid' || $fd eq 'tjid2' || $fd eq 'uid2') {
    next;
  }
  elsif($fd eq 'checkbox') {
    $value = $query->checkbox(-name=>'chkPerson',
      -checked=>($hash{'utid'}=='-1'),
      -value=>('p' . $hash{'personnr'} . 'tj' . $hash{'tjid2'} . 'u'
        . $hash{'uid2'}),
      -label=>'',
      -override=>1);
    if(defined($hash{'uid2'})) {
      push(@uidlst, $hash{'uid2'});
    }
  }
  elsif(defined($hash{$fd})) {
    if($fd eq 'address') {
      $value = qq|<a href="mailto:$hash{$fd}">$hash{$fd}</a>;
    }
    elsif($fd eq 'personnr') {
      $value = qq|<a href="pul.pl\?sid=$sid&txtPersonnr=$hash{$fd}">$hash{$fd}</a>;
    }
    elsif($fd eq 'kid') {
      $value = qq|<a href="
      chledfunk.pl\?sid=$sid&selInstitution=-1&selKstalle=$hash{$fd}&selUppdragstyp=-1"
      >$hash{$fd}</a>;
    }
    else {
      $value = $hash{$fd};
    }
  }
}

```

```

        }
    }
    else {
        $value = '&nbsp;';
    }
    print qq|<td>$value</td>|;
}
print "</tr>\n";
}
print '</table><br />';
print $query->hidden(-name=>'hdnUidLst',
                    -default=>\@uidlst,
                    -override=>1);
}
1;
__END__

```

3.1.3 login.pl

```

use strict 'vars';
use Win32::ODBC;
use CGI;
use CGI::Carp 'fatalsToBrowser';
$CGI::POST_MAX=1024 * 100; # max 100K posts
$CGI::DISABLE_UPLOADS = 1; # no uploads

use Check;

my $db = initDb();

my $query = new CGI;

checkRemoteHost($query);

my $sid;
my $anamn;
my $behorighet = 0;
if($query->param('sid')) {
    $sid = $query->param('sid');

    if($query->param('action')) {
        if($query->param('action') eq 'logout') {
            my $SQLStatement = qq|DELETE FROM Sessioner
                WHERE sid = $sid;|;
            FireSQL($SQLStatement);
        }
        elsif($query->param('action') eq 'login') {
            $anamn = $query->param('anamn');
            my $losenord = $query->param('losenord');
            my $SQLStatement = qq|SELECT aid, behorighet
                FROM Anvandare
                WHERE anamn = '$anamn' AND losenord = '$losenord';|;
            FireSQL($SQLStatement);
            if($db->FetchRow()) {
                my %hash = $db->DataHash;
                my $aid = $hash{'aid'};
                $behorighet = $hash{'behorighet'};
            }
        }
    }
}

```

```

        my ($sec, $min, $hour, $mday, $mon, $year) = localtime();
        my $timenow = ($year+1900) . '-' . ($mon+1) . '-' . $mday . " $hour:$min:$sec";
        $SQLStatement = qq|UPDATE Sessioner
                        SET aid = $aid, senast = '$timenow'
                        WHERE sid = $sid;|;
        FireSQL($SQLStatement);
    }
}
else { # Same as in Check.pm
    my $SQLStatement = qq|SELECT sid, anamn, ip, behorighet
                        FROM Sessioner, Anvandare
                        WHERE Sessioner.aid = Anvandare.aid AND sid = $sid;|;
    FireSQL($SQLStatement);
    if($db->FetchRow()) {
        my %hash = $db->DataHash;
        $anamn = $hash{'anamn'};
        my $ip = $hash{'ip'};
        if($ip eq $query->remote_addr()) {
            $behorighet = $hash{'behorighet'};
        }
        my ($sec, $min, $hour, $mday, $mon, $year) = localtime();
        my $timenow = ($year+1900) . '-' . ($mon+1) . '-' . $mday . " $hour:$min:$sec";
        $SQLStatement = qq|UPDATE Sessioner
                        SET senast = '$timenow'
                        WHERE sid = $sid;|;
        FireSQL($SQLStatement);
    }
}

if($behorighet == 0) {
    my $SQLStatement = qq|SELECT MAX(sid) AS msid
                        FROM Sessioner;|;
    FireSQL($SQLStatement);
    if($db->FetchRow()) {
        my %hash = $db->DataHash;
        $sid = $hash{'msid'};
    }
    $sid = defined($sid) ? $sid : 1;
    $sid += int(rand(100));
    my $ip = $query->remote_addr();
    my ($sec, $min, $hour, $mday, $mon, $year) = localtime();
    my $timenow = ($year+1900) . '-' . ($mon+1) . '-' . $mday . " $hour:$min:$sec";
    my $SQLStatement = qq|INSERT INTO Sessioner (sid, ip, ankom, senast)
                        VALUES ($sid, '$ip', '$timenow', '$timenow');|;
    FireSQL($SQLStatement);
}

#Always close the database connection
$db->Close();

printHead($query, 'Login - Ledfunk', '');

#print $query->Dump;

if($behorighet == 0) {
    print $query->start_form(-name=>'frm',
                            -method=>'post',
                            -action=>'login.pl');
    print $query->hidden(-name=>'sid',
                        -default=>$sid,

```

```

        -override=>1);
print $query->hidden(-name=>'action',
                    -default=>'login',
                    -override=>1);
print "<table><tr><td>Användarnamn:</td><td>";
print $query->textfield(-name=>'anamn',
                      -default=>'',
                      -size=>20,
                      -maxlength=>20);
print "</td></tr><tr><td>Lösenord:</td><td>";
print $query->password_field(-name=>'losenord',
                             -value=>'',
                             -size=>20,
                             -maxlength=>20);

print "</td></tr></table>";
print $query->submit(-name=>'btnSend',
                   -value=>'Skicka');
print $query->endform;
}
else {
print qq|<a href="login.pl\?sid=$sid">Inloggad</a>: $anamn <a
      href="login.pl\?sid=$sid&action=logout">Logga ut</a><br><br>;
if($behorighet >= 1) {
print qq|<a href="pul.pl\?sid=$sid">Visa PuL-utdrag</a><br>;
print qq|<a href="ledfunk.pl\?sid=$sid">Visa ledningsfunktioner</a><br>;
if($behorighet >= 50) {
print qq|<a href="chledfunk.pl\?sid=$sid">Ändra ledningsfunktioner</a><br>;
if($behorighet >= 100) {
print qq|<a href=
      "printcolval.pl\?sid=$sid">Visa tabeller, kolumner och unika värden</a><br>;
print qq|<a href=
      "select.pl\?sid=$sid">Administrera med SQL</a> (Access via ODBC)<br>;
print qq|<a href="formmail.pl\?sid=$sid">Meddela uppdatering</a><br>;
}
}
}
}
print $query->end_html;

__END__

```

3.1.4 pul.pl

```

use strict 'vars';
use Win32::ODBC;
use CGI;
use CGI::Carp 'fatalsToBrowser';
$CGI::POST_MAX=1024 * 100; # max 100K posts
$CGI::DISABLE_UPLOADS = 1; # no uploads

use Check;

my ($db, $query, $sid, $anamn) = check(1);

printHead($query, 'PuL-utdrag - Ledfunk', 'frm.txtPersonnr.focus()');

print qq|<a href="login.pl\?sid=$sid">Inloggad</a>: $anamn
      <a href="login.pl\?sid=$sid&action=logout">Logga ut</a><br><br>;

```

```

print $query->start_form(-name=>'frm',
                        -method=>'post',
                        -action=>'pul.pl');

print $query->hidden(-name=>'sid',
                   -default=>$sid,
                   -override=>1);

print "Personnummer: " . $query->textfield(-name=>'txtPersonnr',
                                           -default=>'',
                                           -size=>11,
                                           -maxlength=>11);

print '&nbsp;'. $query->submit(-name=>'btnSend',
                          -value=>'Skicka');
print $query->endform;

my $SQL;
if($query->param("txtPersonnr")) {
    my $personnr = $query->param("txtPersonnr");
    $personnr =~ s/-//;
    printSQLVt(qq|SELECT personnr, namn, eadress, hamtstalle
              FROM Personer
              WHERE personnr = '$personnr'|, $sid);
    printSQLVt(qq|SELECT telefonnr, tbeskr, Telefoner.kid, knamn, inamn
              FROM ((Telefoner LEFT JOIN Kstallen ON Telefoner.kid = Kstallen.kid)
                   LEFT JOIN Institutioner ON Kstallen.iid = Institutioner.iid)
              WHERE personnr = '$personnr'|, $sid);
    printSQLVt(qq|SELECT btitel, atbeskr, tjfr_o_m, tjt_o_m, omf, tjg_omf, anstnr, Tjanster.kid,
              knamn, inamn
              FROM (((Tjanster INNER JOIN Anst_typer ON Tjanster.atid = Anst_typer.atid)
                   INNER JOIN Befattningar ON Tjanster.bid = Befattningar.bid)
                   INNER JOIN Kstallen ON Tjanster.kid = Kstallen.kid)
              LEFT JOIN Institutioner ON Kstallen.iid = Institutioner.iid)
              WHERE Tjanster.personnr = '$personnr'
              ORDER BY tjid ASC|, $sid);
    printSQLVt(qq|SELECT uttitel, belopp, ufr_o_m, ut_o_m, anstnr, Uppdrag.kid, knamn, inamn
              FROM (((Uppdrag INNER JOIN Uppdragstyper ON Uppdrag.utid = Uppdragstyper.utid)
                   INNER JOIN Kstallen ON Uppdrag.kid = Kstallen.kid)
              LEFT JOIN Institutioner ON Kstallen.iid = Institutioner.iid)
              WHERE Uppdrag.personnr = '$personnr'
              ORDER BY uid ASC|, $sid);
}

#Always close the database connection
$db->Close();

print $query->end_html;

__END__

```

3.1.5 ledfunk.pl

```

use strict 'vars';
use Win32::ODBC;
use CGI;
use CGI::Carp 'fatalsToBrowser';
$CGI::POST_MAX=1024 * 100; # max 100K posts
$CGI::DISABLE_UPLOADS = 1; # no uploads

```

```

use Check;

my ($db, $query, $sid, $anamn) = check(1);

my $SQL;
my $iid = -1;
my $kid = -1;
my $utid = -1;
if($query->param("selInstitution")) {
    ($iid) = $query->param("selInstitution");
    ($kid) = $query->param("selKstalle");
    ($utid) = $query->param("selUppdragstyp");
    $SQL = qq|SELECT DISTINCT inamn, Kstallen.kid, knamn, uttitel, belopp, ufr_o_m, ut_o_m,
        Personer.personnr, namn, eadress, hamtstalle
        FROM Personer, Uppdrag, Institutioner, Kstallen, Uppdragstyper
        WHERE Personer.personnr=Uppdrag.personnr AND Uppdrag.kid=Kstallen.kid
        AND Kstallen.iid=Institutioner.iid AND Uppdrag.utid=Uppdragstyper.utid|;
    if($iid != -1) {
        $SQL .= qq| AND Kstallen.iid=$iid|;
    }
    if($kid != -1) {
        $SQL .= qq| AND Uppdrag.kid=$kid|;
    }
    if($utid != -1) {
        $SQL .= qq| AND Uppdrag.utid=$utid|;
    }
    $SQL .= " ORDER BY Personer.personnr ASC;";
}
else {
    $SQL = "SELECT NULL;";
}

printHead($query, 'Visa - Ledfunk');

#print $behorighet;
#print $query->Dump;
my $hiddensid = $query->hidden(-name=>'sid',
    -default=>$sid,
    -override=>1);

print <<EOF;
<a href="login.pl?sid=$sid">Inloggad</a>: $anamn <a href=
    "login.pl?sid=$sid&action=logout">Logga ut</a><br />
<form method="post" action="ledfunk.pl">
$hiddensid
<table border="0" width="100%" id="table2">
<tr>
<td>Institution:</td>
<td>
EOF

my %inamns; $inamns{'-1'} = 'Alla';
my @iids = '-1';
$db->Sql(qq|SELECT DISTINCT Institutioner.iid, inamn
    FROM Institutioner INNER JOIN
        Kstallen ON Kstallen.iid = Institutioner.iid
    WHERE kid IN (SELECT kid
        FROM Uppdrag)
    ORDER BY inamn ASC;|);
while($db->FetchRow()) {
    my %hash = $db->DataHash();

```



```

    $inamns{$hash{'iid'}} = $hash{'inamn'};
    push(@iids, $hash{'iid'});
}

print $query->scrolling_list(-name=>'selInstitution',
                            -values=>\@iids,
                            -default=>"$iid",
                            -size=>1,
                            -labels=>\%inamns,
                            -override=>1);

print <<EOF;
</td>
</tr>
<tr>
<td>Kställe:</td>
<td>
EOF

my %knamns; $knamns{'-1'} = 'Alla';
my @kids = '-1';

$db->Sql(qq|SELECT DISTINCT kid, knamn
        FROM Kstallen
        WHERE kid IN (SELECT kid
                     FROM Uppdrag)
        ORDER BY knamn ASC|);
while($db->FetchRow()) {
    my %hash = $db->DataHash();
    $knamns{$hash{'kid'}} = $hash{'knamn'};
    push(@kids, $hash{'kid'});
}

print $query->scrolling_list(-name=>'selKstalle',
                            -values=>\@kids,
                            -default=>"$kid",
                            -size=>1,
                            -labels=>\%knamns);

print <<EOF;
</td>
</tr>
<tr>
<td>Ledningsfunktion:</td>
<td>
EOF

my %uttitels; $uttitels{'-1'} = 'Alla';
my @utids = '-1';
$db->Sql('SELECT utid, uttitel FROM Uppdragstyper ORDER BY uttitel ASC');
while($db->FetchRow()) {
    my %hash = $db->DataHash();
    $uttitels{$hash{'utid'}} = $hash{'uttitel'};
    push(@utids, $hash{'utid'});
}

print $query->scrolling_list(-name=>'selUppdragstyp',
                            -values=>\@utids,
                            -default=>"-1",
                            -size=>1,
                            -labels=>\%uttitels);

```

```

print <<EOF;
</td>
</tr>
</table>
<p><input type="submit" value="Skicka" name="btnSendLedfunk"></p>
</form>
<p class="small">$$SQL</p>
EOF

printSQL($SQL, $sid);

#Always close the database connection
$db->Close();

print $query->end_html;

__END__

```

3.1.6 chledfunk.pl

```

use strict 'vars';
use Win32::ODBC;
use CGI;
use CGI::Carp 'fatalsToBrowser';
$CGI::POST_MAX=1024 * 100; # max 100K posts
$CGI::DISABLE_UPLOADS = 1; # no uploads

use Check;

my ($db, $query, $sid, $anamn) = check(50);

my $SQL;
my $iid = -1;
my $kid = -1;
my $utid = -1;
if($kid = $query->param('selKstalle')) {
    $iid = $query->param('selInstitution');
    $kid = $query->param('selKstalle');
    $utid = $query->param('selUppdragstyp');
    my $uppdrag;
    if($utid == -1) {
        $uppdrag = 'U.utid>0';
    }
    else {
        $uppdrag = "U.utid=$utid";
    }
    $SQL = qq|SELECT DISTINCT ($uppdrag AND (Tj.anstnr IS NULL OR Tj.anstnr=U.anstnr)) AS utid,
        namn,
        P.personnr,
        Tj.anstnr,
        IIf(Tj.anstnr IS NULL OR Tj.anstnr=U.anstnr,uttitel,NULL) AS Titel,
        IIf(Tj.anstnr IS NULL OR Tj.anstnr=U.anstnr,belopp,NULL) AS Belopp_,
        IIf(Tj.anstnr>0,tjid,NULL) AS tjid2,
        IIf(Tj.anstnr IS NULL OR Tj.anstnr=U.anstnr,uid,NULL) AS uid2,
        K.kid, K.knamn, I.inamn
    FROM (((Personer AS P LEFT JOIN Tjanster AS Tj ON P.personnr=Tj.personnr) LEFT JOIN
    Uppdrag AS U ON P.personnr=U.personnr) LEFT JOIN
    Uppdragstyper AS UT ON U.utid=UT.utid) LEFT JOIN
    KStallen AS K ON Tj.kid=K.kid) LEFT JOIN

```

```

    Institutioner AS I ON K.iid=I.iid
    WHERE K.kid IS NOT NULL|;
if($kid != -1) {
    $SQL .= qq| AND K.kid=$kid|;
    $iid = -1;
}
elseif($iid != -1) {
    $SQL .= qq| AND K.iid=$iid|;
}
$SQL .= ' ORDER BY P.personnr ASC';
$SQL .= qq| UNION
    SELECT DISTINCT ($uppdrag AND (Tj.anstnr IS NULL OR Tj.anstnr=U.anstnr)) AS utid,
        namn,
        P.personnr,
        Tj.anstnr,
        IIf(Tj.anstnr IS NULL OR Tj.anstnr=U.anstnr,uttitel,NULL) AS Titel,
        IIf(Tj.anstnr IS NULL OR Tj.anstnr=U.anstnr,belopp,NULL) AS Belopp_,
        IIf(Tj.anstnr>0,tjid,NULL) AS tjid2,
        IIf(Tj.anstnr IS NULL OR Tj.anstnr=U.anstnr,uid,NULL) AS uid2,
        K.kid, K.knamn, I.inamn
    FROM (((Personer AS P LEFT JOIN Tjanster AS Tj ON P.personnr=Tj.personnr) LEFT JOIN
    Uppdrag AS U ON P.personnr=U.personnr) LEFT JOIN
    Uppdragstyper AS UT ON U.utid=UT.utid) LEFT JOIN
    KStallen AS K ON U.kid=K.kid) LEFT JOIN
    Institutioner AS I ON K.iid=I.iid
    WHERE K.kid IS NOT NULL|;
if($iid != -1) {
    $SQL .= qq| AND K.iid=$iid|;
}
if($kid != -1) {
    $SQL .= qq| AND K.kid=$kid|;
}
$SQL .= ' ORDER BY P.personnr ASC;';
}
else {
    $SQL = qq|SELECT FALSE AS utid, '' AS namn, '' AS personnr, 0 AS anstnr, 0 AS tjid2,
        0 AS uid2, '' AS Titel, 0 AS Belopp_||;
}

printHead($query, 'Ändra - Ledfunk');

#print $behorighet;
#print $query->Dump;
my $hiddensid = $query->hidden(-name=>'sid',
                             -default=>$sid,
                             -override=>1);

print <<EOF;
<a href="login.pl\?sid=$sid">Inloggad</a>: $namn <a href="login.pl\?sid=$sid&action=logout"
  >Logga ut</a><br>
<form method="post" action="chledfunk.pl">
  $hiddensid
  <table border="0" width="100%" id="table2">
  <tr>
  <td>Institution:</td>
  <td>
  EOF

my %inamns; $inamns{'-1'} = 'Alla';
my @iids = '-1';
$db->Sql(qq|SELECT DISTINCT Institutioner.iid, inamn
        FROM Institutioner INNER JOIN

```

```

        KStallen ON KStallen.iid = Institutioner.iid
        WHERE kid IN (SELECT kid
        FROM Tjanster) OR kid IN (SELECT kid
        FROM Uppdrag)
        ORDER BY inamn ASC;|);
while($db->FetchRow()) {
    my %hash = $db->DataHash();
    $inamns{$hash{'iid'}} = $hash{'inamn'};
    push(@iids, $hash{'iid'});
}

print $query->scrolling_list(-name=>'selInstitution',
                            -values=>\@iids,
                            -default=>"$iid",
                            -size=>1,
                            -labels=>\%inamns,
                            -override=>1);

print <<EOF;
</td>
</tr>
<tr>
<td>Kställe:</td>
<td>
EOF

my %knamns; $knamns{'-1'} = 'Alla';
my @kids = '-1';

$db->Sql(qq|SELECT DISTINCT kid, knamn
        FROM Kstallen
        WHERE kid IN (SELECT kid
        FROM Tjanster) OR kid IN (SELECT kid
        FROM Uppdrag)
        ORDER BY knamn ASC;|);
while($db->FetchRow()) {
    my %hash = $db->DataHash();
    $knamns{$hash{'kid'}} = $hash{'knamn'};
    push(@kids, $hash{'kid'});
}

print $query->scrolling_list(-name=>'selKstalle',
                            -values=>\@kids,
                            -default=>"$kid",
                            -size=>1,
                            -labels=>\%knamns);

print <<EOF;
</td>
</tr>
<tr>
<td>Ledningsfunktion:</td>
<td>
EOF

my %uttitels; $uttitels{'-1'} = 'Alla';
my @utids = '-1';
$db->Sql('SELECT utid, uttitel FROM Uppdragstyper ORDER BY uttitel ASC');
while($db->FetchRow()) {
    my %hash = $db->DataHash();
    $uttitels{$hash{'utid'}} = $hash{'uttitel'};
    push(@utids, $hash{'utid'});
}

```

```

print $query->scrolling_list(-name=>'selUppdragstyp',
                           -values=>\@utids,
                           -default=>"-1",
                           -size=>1,
                           -labels=>\%uttitels);

print <<EOF;
</td>
</tr>
</table>
<p>
EOF

print $query->submit(-name=>'btnSearchLedfunk',
                   -value=>' Sök ');

print <<EOF;
</p>
<p class="small">${SQL}</p>
EOF

if(defined($query->param('btnChangeLedfunk')) {
    $db->Sql('SELECT MAX(uid) AS maxuid FROM Uppdrag');
    my $maxuid;
    while($db->FetchRow()) {
        my %hash = $db->DataHash();
        $maxuid = $hash{'maxuid'};
        ++$maxuid;
    }
    $db->Sql("SELECT MIN(belopp) AS minbelopp FROM Uppdrag WHERE kid=$kid AND utid=$utid");
    my $minbelopp;
    while($db->FetchRow()) {
        my %hash = $db->DataHash();
        $minbelopp = $hash{'minbelopp'};
    }
    if(!defined($minbelopp)) {
        $minbelopp = 'NULL';
    }

    my @chks = $query->param('chkPerson');
    #print '(' . join(', ', @chks) . ')';
    my @personnrns = map(/.*p(\d+)/, @chks);
    #print '(' . join(', ', @personnrns) . ')';
    my @tjids = map(/.*tj(\d+)/, @chks);
    push(@tjids, -999);
    #print '(' . join(', ', @tjids) . ')';
    my @uids = map(/.*u(\d+)/, @chks);
    push(@uids, -999);
    #print '(' . join(', ', @uids) . ')';
    my @uidlst = $query->param('hdnUidLst');
    if($utid != -1 && $kid != -1) {
        my $SQLDelete = qq|DELETE
            FROM Uppdrag
            WHERE Uppdrag.utid=$utid AND Uppdrag.kid=$kid AND uid IN (|
                . join(', ', @uidlst) . ') AND uid NOT IN (|
                . join(', ', @uids) . ')|';
        #print $SQLDelete;
        $db->Sql($SQLDelete);

        my $chk;
        foreach $chk (@chks) {
            $chk =~ m/.*p(\d+)/;

```

```

my $personnr = $1;
$chk =~ m/.*tj(\d+)/;
my $tjid = $1;
$chk =~ m/.*u(\d+)/;
my $uid = $1;
my $SQLInsert =
qq|INSERT INTO Uppdrag (uid, utid, belopp, ufr_o_m, ut_o_m, kid, personnr, anstnr)
  SELECT $maxuid AS uid, $utid AS utid, $minbelopp AS belopp, NULL AS ufr_o_m,
    NULL AS ut_o_m, $kid AS kid, Personer.personnr, Tjanster.anstnr
  FROM ((Personer LEFT JOIN Tjanster
    ON Personer.personnr=Tjanster.personnr) LEFT JOIN Uppdrag
    ON Personer.personnr=Uppdrag.personnr) LEFT JOIN Uppdragstyper
    ON Uppdrag.utid=Uppdragstyper.utid
  WHERE Personer.personnr = '$personnr'
    AND Personer.personnr NOT IN (SELECT personnr
      FROM Uppdrag
      WHERE utid=$utid AND kid=$kid
      AND Tjanster.anstnr IS NULL
      OR Uppdrag.anstnr=Tjanster.anstnr)
    AND (Tjanster.anstnr IS NULL OR Tjanster.anstnr IN (SELECT anstnr
      FROM Tjanster
      WHERE tjid=$tjid));|;

#print $SQLInsert;
#printSQL($SQLInsert, $sid);
$db->Sql($SQLInsert);
++$maxuid;
}
}
else {
  print 'Du måste välja kställe och ledningsfunktion för att kunna ändra.';
}
}

printSQLCheckbox($SQL, $sid);
print $query->submit(-name=>'btnChangeLedfunk',
  -value=>'Ändra');
print ' ' . $query->reset(-name=>'btnResetLedfunk',
  -value=>'Återställ');
print '</form>';

#Always close the database connection
$db->Close();

print $query->end_html;

__END__

```

3.1.7 printcolval.pl

```

use strict 'vars';
use Win32::ODBC;
use CGI;
use CGI::Carp 'fatalsToBrowser';
$CGI::POST_MAX=1024 * 100; # max 100K posts
$CGI::DISABLE_UPLOADS = 1; # no uploads

use Check;

my ($db, $query, $sid, $anamn) = check(100);

```

```

printHead($query, 'Tabeller, kolumner och unika värden - Ledfunk');

print qq|<a href="login.pl?sid=$sid">Inloggad</a>: $anamn <a
      href="login.pl?sid=$sid&action=logout">Logga ut</a><br><br>;

my @tables = $db->TableList;

my $table;
foreach $table (sort @tables) {
  if($table =~ /^[_a-zA-Z]/) {
    $db->Sql("SELECT * FROM \[$table\]");
    $db->FetchRow();
    my @FieldNames = $db->FieldNames();
    my $fd;
    foreach $fd (@FieldNames) {
      print "<b>$table.$fd</b>";
      my $SQL = qq|SELECT DISTINCT \[$fd\]
                FROM \[$table\]|;
      FireSQL($SQL);
      my $rows = 0;
      while($db->FetchRow()) {
        ++$rows;
      }
      print "$rows;<BR>\n";
      if($rows < 70 && !($fd =~ /dat/)) {
        $db->Sql($SQL);
        while($db->FetchRow()) {
          my %hash = $db->DataHash;
          print "  $hash{$fd};<br>\n";
        }
        print "<br>\n";
      }
    }
    print "<br>\n";
  }
}

#Always close the database connection
$db->Close();

print $query->end_html;

__END__

```

3.1.8 select.pl

```

use strict 'vars';
use Win32::ODBC;
use CGI;
use CGI::Carp 'fatalsToBrowser';
$CGI::POST_MAX=1024 * 100; # max 100K posts
$CGI::DISABLE_UPLOADS = 1; # no uploads

use Check;

my ($db, $query, $sid, $anamn) = check(100);

my $SQL;

```

```

if($query->param("taSQL")) {
    $SQL = $query->param("taSQL");
}
else {
    $SQL = "SELECT NULL;";
}

printHead($query, 'SQL - Ledfunk', 'frm.taSQL.focus()');

#print $behorighet;
#print $query->Dump;

my $hiddensid = $query->hidden(-name=>'sid',
                             -default=>$sid,
                             -override=>1);

print <<EOF;
<a href="login.pl\?sid=$sid">Inloggad</a>: $anamm <a
    href="login.pl\?sid=$sid&action=logout">Logga ut</a><br /><br />
EOF

print $query->start_form(-name=>'frm',
                       -method=>'post',
                       -action=>'select.pl');

print $hiddensid;
print "Skriv din SQL-fråga för Access via ODBC här:<br />";
print $query->textarea(-name=>'taSQL',
                     -default=>$SQL,
                     -rows=>5,
                     -columns=>80);

print $query->br;
print $query->submit(-name=>'btnSend',
                   -value=>'Skicka');

print $query->br;
print $query->endform;

printSQL($SQL, $sid);

#Always close the database connection
$db->Close();

print $query->end_html;

__END__

```

3.1.9 formmail.pl

```

use strict;
use Mail::Sender;
use Win32::ODBC;
use CGI;
use CGI::Carp 'fatalsToBrowser';
$CGI::POST_MAX=1024 * 100; # max 100K posts
$CGI::DISABLE_UPLOADS = 1; # no uploads

use Check;

my ($db, $query, $sid, $anamm) = check(100);

```



```

printHead($query, 'Uppdateringsbrev - Ledfunk', 'frm.taBody.focus()');

#print $behorighet;
#print $query->Dump;
my $hiddensid = $query->hidden(-name=>'sid',
                             -default=>$sid,
                             -override=>1);

print <<EOF;
<a href="login.pl\?sid=$sid">Inloggad</a>: $anamm <a
  href="login.pl\?sid=$sid&action=logout">Logga ut</a><br /><br />
EOF

if(!defined $query->param('taBody')) {
  print $query->start_form(-name=>'frm',
                          -method=>'post',
                          -action=>'formmail.pl');

  print $hiddensid;
  print "Skriv brev om hur databasen har uppdaterats:<br>";
  print $query->textarea(-name=>'taBody',
                        -default=>' ',
                        -rows=>10,
                        -columns=>50);

  print $query->br;
  print $query->submit(-name=>'btnSend',
                     -value=>'Skicka');

  print $query->br;
  print $query->endform;
}
else {
  my $sender = new Mail::Sender {
    smtp => 'smtp.lu.se',
    from => $Check::from,
    on_errors => undef}
    or die "Can't create the Mail::Sender object: $Mail::Sender::Error\n";
  $sender->MailMsg({
    to => $Check::to,
    subject => 'Uppdatering av Ledfunk-databasen har skett',
    msg => "Brevet skickades: " . scalar localtime . ".\n"
      . "IP-nummer: " . $query->remote_host() . ".\n"
      . "Användarnamn: " . $anamm . ".\n"
      . "Sessionsid: " . $sid . ".\n\n"
      . $query->param('taBody')
  });
  if($sender->{'error'} < 0) {
    print "ERROR: $Mail::Sender::Error\n";
  }
  else {
    print "Meddelandet har skickats.\n";
  }
}

#Always close the database connection
$db->Close();

print $query->end_html;

__END__

```

3.2 Skapa Ledfunk-databasen

3.2.1 Common.pm

I denna Perl-modul finns funktioner som används av alla andra Perl-program.

```
$Common::db = NULL;

sub initDb {
    #Define the Data Source Name
    my $DSN = "ledfunk";

    #Create a Win32::ODBC object
    $Common::db = new Win32::ODBC($DSN);

    #Verify the connection is valid
    if(!$Common::db) {
        print "Failed to Connect $DSN\n";
        Win32::ODBC::DumpError();
        #You can't do anything without a connection so die here
        die;
    }
    return $Common::db;
}

sub FireSQL {
    my $SQL = shift @_;
    if($Common::db->Sql($SQL) {
        print "Error running statement $SQL";
        Win32::ODBC->DumpError();
    }
}

sub FireSQLRI {
    my $SQL = shift @_;
    print "Begin query.\n";
    if($Common::db->Sql($SQL) {
        print "Error running statement $SQL";
        Win32::ODBC->DumpError();
    }
    else {
        print $Common::db->RowCount($Common::db->Connection()) . " rows inserted.\n";
    }
}

1;

__END__
```

3.2.2 createdsn.pl

```
use Cwd;
use Win32::ODBC;

use Common;

#Define the driver type for this database
my $DriverType = "Microsoft Access Driver (*.mdb)";
#Define the Data Source Name
```

```

my $DSN = "ledfunk";
#Describe the Data Source Name
my $Description = "Description=ledfunk";
#The filename of the database.
#This must be created before you can connect to the DSN
my $DataBase = "ledfunk.mdb";
#Set the directory to current directory
my $dir = cwd();
#Configure the DSN
if(Win32::ODBC::ConfigDSN(ODBC_ADD_DSN,
                          $DriverType,
                          ("DSN=$DSN",
                           $Description,
                           "DBQ=$dir\\$DataBase",
                           "DEFAULTDIR=$dir",
                           "UID=", "PWD="))) {
    print "Successful configuration of $DSN!\n";
}
else {
    print "Error Creating $DSN\n";
    #Always use the DumpError routine.
    #It tells you what is going on
    Win32::ODBC::DumpError();
    die;
}

my $db = initDb();

my $conn = $db->Connection();
print "Successful Connection $conn, $DSN\n";

#Always close the database connection
$db->Close();

__END__

```

3.2.3 createtables.pl

```

use strict 'vars';
use Win32::ODBC;

use Common;

my $db = initDb();

my $SQLStatement = qq|CREATE TABLE Institutioner
    (iid INTEGER NOT NULL,
     inamn VARCHAR(50),
     omrade VARCHAR(10),
     PRIMARY KEY (iid));|;

FireSQL($SQLStatement);

$SQLStatement = qq|CREATE TABLE Kstallen
    (kid INTEGER NOT NULL,
     knamn VARCHAR(50),
     iid INTEGER,
     PRIMARY KEY (kid),
     FOREIGN KEY (iid) REFERENCES Institutioner);|;

```

```
FireSQL($SQLStatement);

$SQLStatement = qq|CREATE TABLE Personer
  (personnr VARCHAR(10) NOT NULL,
   namm VARCHAR(50),
   fodd DATE,
   eadress VARCHAR(50),
   hamtstalle INTEGER,
   PRIMARY KEY (personnr));|;

FireSQL($SQLStatement);

$SQLStatement = qq|CREATE TABLE Telefoner
  (telefonnr VARCHAR(20) NOT NULL,
   tbeskr VARCHAR(50),
   personnr VARCHAR(10) NOT NULL,
   kid INTEGER NOT NULL,
   PRIMARY KEY (telefonnr),
   FOREIGN KEY (personnr) REFERENCES Personer,
   FOREIGN KEY (kid) REFERENCES Kstallen);|;

FireSQL($SQLStatement);

$SQLStatement = qq|CREATE TABLE Anst_typer
  (atid INTEGER NOT NULL,
   atbeskr VARCHAR(50),
   PRIMARY KEY (atid));|;

FireSQL($SQLStatement);

$SQLStatement = qq|CREATE TABLE Befattningar
  (bid INTEGER NOT NULL,
   btitel VARCHAR(50),
   PRIMARY KEY (bid));|;

FireSQL($SQLStatement);

$SQLStatement = qq|CREATE TABLE Tjanster
  (tjid INTEGER NOT NULL,
   bid INTEGER NOT NULL,
   atid INTEGER NOT NULL,
   kid INTEGER NOT NULL,
   personnr VARCHAR(10) NOT NULL,
   anstnr INTEGER NOT NULL,
   tjfr_o_m DATE,
   tjt_o_m DATE,
   omf DOUBLE,
   tjg_omf DOUBLE,
   PRIMARY KEY (tjid),
   FOREIGN KEY (bid) REFERENCES Befattningar,
   FOREIGN KEY (atid) REFERENCES Anst_typer,
   FOREIGN KEY (kid) REFERENCES Kstallen,
   FOREIGN KEY (personnr) REFERENCES Personer);|;

FireSQL($SQLStatement);

$SQLStatement = qq|CREATE TABLE Uppdragstyper
  (utid INTEGER NOT NULL,
   uttitel VARCHAR(50),
   PRIMARY KEY (utid));|;
```

```

FireSQL($SQLStatement);

$SQLStatement = qq|CREATE TABLE Uppdrag
  (uid INTEGER NOT NULL,
  utid INTEGER NOT NULL,
  belopp DOUBLE,
  ufr_o_m DATE,
  ut_o_m DATE,
  kid INTEGER NOT NULL,
  personnr VARCHAR(10) NOT NULL,
  anstnr INTEGER NOT NULL,
  PRIMARY KEY (uid),
  FOREIGN KEY (utid) REFERENCES Uppdragstyper,
  FOREIGN KEY (kid) REFERENCES Kstallen,
  FOREIGN KEY (personnr) REFERENCES Personer);|;

#FOREIGN KEY (personnr, anstnr) REFERENCES Tjanster

FireSQL($SQLStatement);

$db->Close();

__END__

```

3.2.4 insert.pl

```

use strict 'vars';
use Win32::ODBC;

use Common;

my $db = initDb();

my $SQLStatement;

# kstalle och institution har bytt betydelse

$SQLStatement = qq|INSERT INTO Institutioner (iid, inamn, omrade)
  SELECT val(left(Instlista_0410.kstalle, 6)), mid(Instlista_0410.kstalle, 8), omrade
  FROM Instlista_0410 LEFT JOIN Anst0410
  ON val(left(Instlista_0410.institution, 6)) = val(Anst0410.kstallenr)
  WHERE NOT omrade IS NULL
  GROUP BY Instlista_0410.kstalle, omrade;|;

FireSQLRI($SQLStatement);

$SQLStatement = qq|INSERT INTO Institutioner (iid, inamn, omrade)
  SELECT val(left(Instlista_0410.kstalle, 6)), mid(Instlista_0410.kstalle, 8), omrade
  FROM Instlista_0410 LEFT JOIN Anst0410
  ON val(left(Instlista_0410.institution, 6)) = val(Anst0410.kstallenr)
  WHERE omrade IS NULL AND val(left(Instlista_0410.kstalle, 6)) NOT IN (
  SELECT DISTINCT val(left(Instlista_0410.kstalle, 6))
  FROM Instlista_0410 LEFT JOIN Anst0410
  ON val(left(Instlista_0410.institution, 6)) = val(Anst0410.kstallenr)
  WHERE NOT omrade IS NULL)
  GROUP BY Instlista_0410.kstalle, omrade;|;

```

```
FireSQLRI($SQLStatement);

$SQLStatement = qq|INSERT INTO Kstallen (kid, knamn, iid)
  SELECT val(left(institution, 6)), mid(institution, 8), val(left(kstalle, 6))
  FROM Instlista_0410
  GROUP BY institution, kstalle;|;

FireSQLRI($SQLStatement);

$SQLStatement = qq|INSERT INTO Kstallen (kid, knamn, iid)
  SELECT DISTINCT val(left(institution, 6)), mid(institution, 8), NULL
  FROM Anst0410
  WHERE val(left(institution, 6)) NOT IN (SELECT val(left(Instlista_0410.institution, 6))
  FROM Instlista_0410);|;

FireSQLRI($SQLStatement);

$SQLStatement = qq|INSERT INTO Kstallen (kid, knamn, iid)
  VALUES (888888, 'Systemkod', NULL);|;

FireSQLRI($SQLStatement);

$SQLStatement = qq|INSERT INTO Kstallen (kid, knamn, iid)
  VALUES (0, 'Okant', NULL);|;

FireSQLRI($SQLStatement);

$SQLStatement = qq|INSERT INTO Uppdragstyper (utid, uttitel)
  SELECT DISTINCT val(left(loneart, 4)), mid(loneart, 6)
  FROM Utlg_pri0410;|;

FireSQLRI($SQLStatement);

$SQLStatement = qq|INSERT INTO Anst_typer (atid, atbeskr)
  SELECT DISTINCT 1000000*len(anstTyp)+1000*asc(left(anstTyp,1))+asc(right(anstTyp,1)), anstTyp
  FROM Anst0410;|;

FireSQLRI($SQLStatement);

$SQLStatement = qq|INSERT INTO Personer (personnr, namn, fodd, eadress, hamtstalle)
  SELECT DISTINCT Anst0410.personnr, namn, '19' & mid(Anst0410.personnr,1,2) & '-'
    & mid(Anst0410.personnr,3,2) & '-' & mid(Anst0410.personnr,5,2), [e-post], hs
  FROM Anst0410 LEFT JOIN [e-post_prefekt041008]
  ON val(Anst0410.personnr) = [e-post_prefekt041008].personnr
  WHERE val(mid(Anst0410.personnr,5,2)) <= 31;|;

FireSQLRI($SQLStatement);

$SQLStatement = qq|INSERT INTO Personer (personnr, namn, fodd, eadress, hamtstalle)
  SELECT DISTINCT Anst0410.personnr, namn, '19' & mid(Anst0410.personnr,1,2) & '-'
    & mid(Anst0410.personnr,3,2) & '-' & val(mid(Anst0410.personnr,5,2))-60, [e-post], hs
  FROM Anst0410 LEFT JOIN [e-post_prefekt041008]
  ON val(Anst0410.personnr) = [e-post_prefekt041008].personnr
  WHERE val(mid(Anst0410.personnr,5,2)) >= 60;|;

FireSQLRI($SQLStatement);

$SQLStatement = qq|INSERT INTO Personer (personnr, namn, fodd, eadress, hamtstalle)
  SELECT DISTINCT left(Utlg_pri0410.personnr, 6) & mid(Utlg_pri0410.personnr, 8), namn, '19'
    & mid(Utlg_pri0410.personnr,1,2) & '-' & mid(Utlg_pri0410.personnr,3,2) & '-'
    & mid(Utlg_pri0410.personnr,5,2), [e-post], hs
  FROM Utlg_pri0410 LEFT JOIN [e-post_prefekt041008]
```

```

ON val(left(Utlg_pri0410.personnr, 6) & mid(Utlg_pri0410.personnr, 8))
    = [e-post_prefekt041008].personnr
WHERE left(Utlg_pri0410.personnr, 6) & mid(Utlg_pri0410.personnr, 8)
    NOT IN (SELECT DISTINCT personnr
FROM Anst0410);|;

FireSQLRI($SQLStatement);

$SQLStatement = qq|INSERT INTO Befattningar (bid, btitel)
    SELECT MAX(Nr), befattning
    FROM Anst0410
    GROUP BY befattning;|;

FireSQLRI($SQLStatement);

$SQLStatement = qq|INSERT INTO Tjanster (tjid, bid, atid, kid, personnr, anstnr, tjfr_o_m,
    tjt_o_m, omf, tjc_omf)
    SELECT nr, bid, atid, kstallenr, personnr, a, NULL, '20' & mid(tom,1,2) & '-' & mid(tom,4,2)
    & '-' & mid(tom,7,2), omf, [tjc omf]
    FROM Anst0410, Befattningar, Anst_typer
    WHERE Anst0410.befattning = Befattningar.btitel AND Anst0410.ansttyp = Anst_typer.atbeskr
    AND len(tom) = 8;|;

FireSQLRI($SQLStatement);

$SQLStatement = qq|INSERT INTO Tjanster (tjid, bid, atid, kid, personnr, anstnr, tjfr_o_m, tjt_o_m,
    omf, tjc_omf)
    SELECT nr, bid, atid, kstallenr, personnr, a, NULL, NULL, omf, [tjc omf]
    FROM Anst0410, Befattningar, Anst_typer
    WHERE Anst0410.befattning = Befattningar.btitel AND Anst0410.ansttyp = Anst_typer.atbeskr
    AND tom IS NULL;|;

FireSQLRI($SQLStatement);

$SQLStatement = qq|INSERT INTO Uppdrag (uid, utid, belopp, ufr_o_m, ut_o_m, kid, personnr, anstnr)
    SELECT nr, utid, belopp, NULL, NULL, val(left(institution, 6)), left(Utlg_pri0410.personnr, 6) &
    mid(Utlg_pri0410.personnr, 8), a
    FROM Utlg_pri0410, Uppdragstyper
    WHERE mid(Utlg_pri0410.loneart,6) = Uppdragstyper.uttitel AND a IS NOT NULL;|;

FireSQLRI($SQLStatement);

$SQLStatement = qq|INSERT INTO Uppdrag (uid, utid, belopp, ufr_o_m, ut_o_m, kid, personnr, anstnr)
    SELECT nr, utid, belopp, NULL, NULL, 0, left(Utlg_pri0410.personnr, 6)
    & mid(Utlg_pri0410.personnr, 8), 1
    FROM Utlg_pri0410, Uppdragstyper
    WHERE mid(Utlg_pri0410.loneart,6) = Uppdragstyper.uttitel AND Utlg_pri0410.nr NOT IN (SELECT uid
    FROM Uppdrag);|;

FireSQLRI($SQLStatement);

$db->Close();

__END__

```

3.2.5 droptables.pl

```

use strict 'vars';
use Win32::ODBC;

```

```
use Common;

my $db = initDb();

my $SQLStatement;

$SQLStatement = qq|DROP TABLE Uppdrag;|;
FireSQL($SQLStatement);

$SQLStatement = qq|DROP TABLE Uppdragstyper;|;
FireSQL($SQLStatement);

$SQLStatement = qq|DROP TABLE Tjanster;|;
FireSQL($SQLStatement);

$SQLStatement = qq|DROP TABLE Befattningar;|;
FireSQL($SQLStatement);

$SQLStatement = qq|DROP TABLE Anst_typer;|;
FireSQL($SQLStatement);

$SQLStatement = qq|DROP TABLE Telefoner;|;
FireSQL($SQLStatement);

$SQLStatement = qq|DROP TABLE Personer;|;
FireSQL($SQLStatement);

$SQLStatement = qq|DROP TABLE Kstallen;|;
FireSQL($SQLStatement);

$SQLStatement = qq|DROP TABLE Institutioner;|;
FireSQL($SQLStatement);

$db->Close();

__END__
```

3.2.6 createheadingstable.pl

```
use strict 'vars';
use Win32::ODBC;

use Common;

my $db = initDb();

my $SQLStatement = qq|CREATE TABLE Rubriker
    (rid VARCHAR(10) NOT NULL,
```



```
        rnamn VARCHAR(20),
        PRIMARY KEY (rid));|;

FireSQL($SQLStatement);

$db->Close();

__END__
```

3.2.7 insertheadings.pl

```
use strict 'vars';
use Win32::ODBC;

use Common;

my $db = initDb();

my @tables = $db->TableList;

my $table;
foreach $table (sort @tables) {
    if($table =~ /^[A-Z]/) {
        $db->Sql("SELECT * FROM \"[$table]");
        $db->FetchRow();
        my @fieldNames = $db->FieldNames();
        my $fd;
        foreach $fd (@fieldNames) {
            if($fd =~ /^[a-z]/) {
                my $SQLStatement;
                $SQLStatement = qq|INSERT INTO Rubriker (rid)
                    VALUES ('$fd')|;
                FireSQLRI($SQLStatement);
            }
        }
    }
}

$db->Close();

__END__
```

3.2.8 dropheadingstable.pl

```
use strict 'vars';
use Win32::ODBC;

use Common;

my $db = initDb();

my $SQLStatement;

$SQLStatement = qq|DROP TABLE Rubriker|;

FireSQL($SQLStatement);
```

```
$db->Close();  
  
__END__
```

3.3 Skapa sessionsdatabasen

3.3.1 createsessiontables.pl

```
use strict 'vars';  
use Win32::ODBC;  
  
use Common;  
  
my $db = initDb();  
  
my $SQLStatement = qq|CREATE TABLE Anvandare  
  (aid INTEGER NOT NULL,  
   anamn VARCHAR(20) NOT NULL,  
   losenord VARCHAR(20) NOT NULL,  
   behorighet INTEGER NOT NULL,  
   PRIMARY KEY (aid));|;  
  
FireSQL($SQLStatement);  
  
$SQLStatement = qq|CREATE TABLE Sessioner  
  (sid INTEGER NOT NULL,  
   ip VARCHAR(15) NOT NULL,  
   aid INTEGER,  
   ankom DATETIME,  
   senast DATETIME,  
   PRIMARY KEY (sid),  
   FOREIGN KEY (aid) REFERENCES Anvandare);|;  
  
FireSQL($SQLStatement);  
  
$db->Close();  
  
__END__
```

3.3.2 dropsessiontables.pl

```
use strict 'vars';  
use Win32::ODBC;  
  
use Common;  
  
my $db = initDb();  
  
my $SQLStatement;  
  
$SQLStatement = qq|DROP TABLE Sessioner;|;  
  
FireSQL($SQLStatement);  
  
$SQLStatement = qq|DROP TABLE Anvandare;|;
```

```
FireSQL($SQLStatement);
```

```
$db->Close();
```

```
__END__
```