# Applikation PHP

Erik Gustafsson (a23erigu)

Table of Contents

[Applikation PHP 1](#_Toc179040510)

[Intro 2](#_Toc179040511)

[Länk till websidan 2](#_Toc179040512)

[Lägga till data i tabell 3](#_Toc179040513)

[Listbox (dropbox) genererad från data 5](#_Toc179040514)

## Intro

Denna rapport är skapad för inlämningen applikation PHP från kursen Databaskonstruktion G1F.   
Inlämningen handlar om en hemsida som skapats åt en databas med hur denna hemsida är kopplad till och påverkar databasen.   
Rapporten kommer främst fokusera på hur lösningar till problem som uppgiften bet lösas.  
Endast php och html har använts till detta projekt och till php användes PDO för att koppla till databasen.  
VG försöktes nå med detta program så de uppgifterna kommer tas upp i slutet.

## Länk till websidan

<https://wwwlab.webug.se/databaskonstruktion/a23erigu/Inlogg.php>

Denna länk tar en till inloggning sidan där man loggar in och sedan tar sig till huvudsidan TomteVerkstad.php.

Användarna som finns är:  
 ”dbkonstruktion” med lösenord ”Skata#23” (root användare).  
” a23eriguByggarNisse” med lösenord ”ByggaBil” (kan vara så att de förlorat rättigheter).

## Lägga till data i tabeller

Första delen som kommer gås över är hur information läggs till i databasen.  
Resultatet av förändringen ska gå att se i sökningen av tomtenissar (namnet på den nya tomtenissen ska vara i sökfältet direkt efter skapandet så bara att klicka på sökknappen).

|  |
| --- |
| <form action = "TomteVerkstad.php" method = "POST">  <p> Create tomtenisse </p>  Name of tomtenisse  <input type="text" name="Name" value=""> <br>  ID of tomtenisse  <input type="text" name="CId" value=""> <br>  Amount of nuts earned  <input type="number" name="Nuts" value=""> <br>  Amount of raisin earned  <input type="number" name="Raisin" value=""> <br>  <input type="submit" name="submit" value="Submit">  </form> |

Figur 1: html för infogandet av data.

Denna kod snut är i html och finns för att kunna ta in data som behövs för att kunna skapa Tomtenissar. Alla fält måste ha något värde för att en tomtenisse ska kunna skapas och ID måste vara i formatet 000000-0000-0-000000000.   
POST används för alla tillfällen där information ska tas med i programmet (med ett undantag) för att kunna uppnå lite bättre säkerhet.

|  |
| --- |
| If (isset($\_POST["Name"]) && isset($\_POST["CId"]) && isset($\_POST["Nuts"]) && isset($\_POST["Raisin"])) {  $CreateTomtenissar = $pdo->prepare("insert into Tomtenisse(Namn, IdNr, Nötter, Russin) values(?, ?, ?, ?)");  $CreateTomtenissar->bindParam(1, $\_POST["Name"],  PDO::PARAM\_STR);  $CreateTomtenissar->bindParam(2, $\_POST["CId"], PDO::PARAM\_STR);  $CreateTomtenissar->bindParam(3, $\_POST["Nuts"], PDO::PARAM\_INT);  $CreateTomtenissar->bindParam(4, $\_POST["Raisin"], PDO::PARAM\_INT);  $CreateTomtenissar->execute(); |

Figure 2: säkerhets koll och infogande i databasen.

Här är php koden och den börjar med att kolla att alla POST måste vara med för att kunna köras.   
Om all information är med så kör det sedan en prepare på en insert i databasen. Frågetecknen i inserten byts sen ut med hjälp av bindparamater till de informationen de ska innehålla. Till sista så används execute för att köra inserten.  
Anledningen prepare används är för att stoppa SQL injektions från att kunna hända.

|  |
| --- |
| if (($CreateTomtenissar->rowCount()) > 0){  echo "<br>";  echo("Insert successful");  }  else{  echo "<br>";  echo("Insert failed");  echo "<br>";  echo $CreateTomtenissar->errorCode();  echo "<br>";  print\_r($CreateTomtenissar->errorInfo());  } |

Figur 3: check på infogandet av Tomtenissar.

Sista delen av skapandet är en check på om det lyckades skapa en Tomtenisse genom att kolla om inserten påverka någon rad i databasen. Om det misslyckades så kommer programmet att skriva ut ett error meddelande med hjälp av errorCode och errorInfo vilket skriver ut error koden samt info om errorer.

## Dropdown (Listbox) genererad från data

Dropdown menyn använda i programmet för att vissa vad namnet på alla Tomtenissar är så man vet vad som går att söka på.

|  |
| --- |
| $Dropdownissar = $pdo->prepare("CALL getNissar");  $Dropdownissar->execute();  $FetchDropdownissar = $Dropdownissar->fetchAll(PDO::FETCH\_ASSOC);  $Dropdownissar->closeCursor();  echo "<p> Name of all tomtenissar </p>";  echo "<select name = 'dropdown'>";  foreach($FetchDropdownissar as $row) {  echo("<option>" . $row["Namn"] . "</option>");  }  echo "</select>"; |

Figur 4: koden för dropdown.

Denna del av koden behöver inte ta någon information från användaren utan ska alltid köras när hemsidan hämtas.

Koden börjar med att göra en prepare statment på stored proceduren ”CALL getNissar”, detta hämtar alla nissar som finns och ska kunna användas av alla användare (kommer gå mer in i det i exekvera en procedur delen). Efter att ha hämtat all information så görs en foreach för att gå igenom varje rad och skriva namnet på användare i dropdown menyn. Dropbox menyn skapas genom "<select name = 'dropdown'>” vilket skapar skjälvaste menyn medans i loopen så säts ”<option>” runt varje rad vilket skapar raderna i dropboxen.

## Sökning i databasen

Sökningen använda för att kunna hitta information om någon tomtenisse genom deras namn.

|  |
| --- |
| <?php  if (empty($\_POST["Name"])) {  $name = "Name";  }  else {  $name = $\_POST["Name"];  }  ?>  <form action = "TomteVerkstad.php" method = "POST">  <p> Search for Tomtenissar </p>  <input type="text" name="Name" value= <?php echo $name ?>>  <input type="submit" name="submit" value="Submit">  </form> |

Figur 5: hämta namn för sökningen.

Den första delen av koden är php och kollar om de fåt in något namn. Har de inte det så sätter programmet namnet till ”Name”.   
De andra delen av koden är html och skapar rutan där användare kan skriva in information. De unika här är inputen Name vilket har sitt värde sätt till namnet som användaren senast skrev in vilket göt att de sparar vad användaren senast sökt på. Name fältet byts också ut till namnet på nya tomtenissar om det skapats en och till tomtenissens namn om det ändrats och programmet ska ta fram de nya/ändrade informationen (för någon anledning dock så blir det den gammal informationen som visas så användaren får trycka på sök knapen igen för att få korrekt information).

|  |
| --- |
| if(isset($\_POST["Name"])){  $Tomtenissar = $pdo->prepare("select \* from Tomtenisse where Namn = ?");  $Tomtenissar->bindParam(1, $name, PDO::PARAM\_STR);  $Tomtenissar->execute();  } |

  Figur 6: prepare statement för att hämta tomtenissar.

Här så skapas prepare statmentet för att kunna få ut informationen ur databasen och bind parameten för att sätta in namnet.

|  |
| --- |
| if (($Tomtenissar->rowCount()) > 0){  foreach($Tomtenissar as $row) {  echo("<pre>");  print\_r($row);  echo("</pre>");  echo "<a href='https://wwwlab.webug.se/databaskonstruktion/a23erigu/TomteVerkstad.php?Name=" . $row["Namn"] . "&Id=" . $row["IdNr"] . "'> Delete " . $row["Namn"] . " </a>";  }  }  else{  echo "<br>";  echo("No info found");  } |

Figur 7: skriva ut de hittade nissarna.

Sista delen av utskrivningen av tomtenissar börjar med att kolla om de får ut någon information. Om det får någon information så körs en loop för att skriva ut de tomtenissarna genom ” print\_r($row)”.   
”<pre>” vilket skrivs ut innan och efter varje tomtenisse finns för att göra så informationen från tomtenissarna är enklare att läsa.   
De sista som skrivs ut åt varje nisse är en länk med nissens namn vilket tar bort nissen om klickad på (denna länk pratas mer om under Generera länkar delen).