

Database-baseret Web-publicering, efterår 2001

Forelæsning 12, tirsdag den 20. november 2001.

PHP3, MySQL og Generisk Login mekanisme

- MySQL (datamodel pladekartotek)
- PHP3
- Pladekartotek
- Hvad er en Cookie
- Problemer med Cookies
- Adgangskontrol i AOLserver
- Lognimekanisme

MySQL

- MySQL: <http://www.mysql.com>
- MySQL Tutorial: <http://www.it-c.dk/~jcjg/IP/F2001/Ugesedler/mysql.html>.

MySQL er bl.a. installeret på maskinen `mysql.itu.dk`

Referencemanual:

`http://mysql.it-c.dk:8080/~henrik/courses/IP/mysql/manual_toc.html`

MySQL-klient (programmet `mysql`)

Man forbinder til databasen med klienten `mysql`:

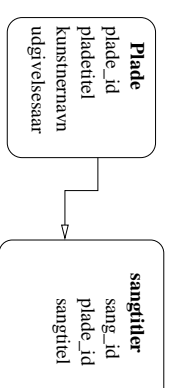
```
[nh@mysql Pladekartotek]$ mysql dwp -h mysql.itu.dk -u nh -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8561 to server version: 3.22.32

Type 'help' for help.
```

hvor `dwp` er navnet på databasen, `mysql.itu.dk` er maskinen hvor databasen kører, `nh` er brugeren der logger på og `-p` betyder at der skal indtastes et lösen for at logge på.

Oprettelse af tabeller i pladekartoteket (MySQL, fil `pladekartotek.mysql`)

MySQL anvender sin egen version af SQL som tildels ligner Oracles SQL-varianter.



I denne løsning anvendes et unikt nummer `sang_id` til hver sang, hvorfor `sang_id` kan udgøre primærnøglen selv.

Plene angiver en-til-mange relationerne, dvs. en plade kan have tilknyttet nul, en eller flere sange.

```
create table plade (
  plade_id integer not null auto_increment,
  pladetitel varchar(100),
  kunstnernavn varchar(100),
  udgivelsesaar integer,
  primary key(plade_id)
);

create table sangtitler (
  plade_id integer not null,
  sang_id integer not null auto_increment,
  sangtitel varchar(100),
  primary key (sang_id)
);
```

Største forskel er `auto_increment` i stedet for `sekvenser`.

Indsættelse af rækker i pladekartoteket (MySQL)

```
insert into plade (pladetitel, kunstnernavn, udgivelsesaar)
  values ('Pladetitel 1', 'Kunstner 1', 1999);
insert into sangtitler (plade_id, sangtitel) values (1, 'Sangtitel 1');
insert into sangtitler (plade_id, sangtitel) values (1, 'Sangtitel 2');

insert into plade (pladetitel, kunstnernavn, udgivelsesaar)
  values ('Pladetitel 2', 'Kunstner 2', 2000);
insert into sangtitler (plade_id, sangtitel) values (2, 'Sangtitel 3');
insert into sangtitler (plade_id, sangtitel) values (2, 'Sangtitel 4');
```

Vi angiver ikke nogen værdi for henholdsvis `plade_id` i tabellen `plade` samt `sang_id` i tabellen `sangtitler`.

Attributen `auto_increment` finder automatisk det næste nummer i rækken – startende fra 1.

Select formaterer uddata pænere i MySQL end i Oracle

```
mysql> select * from plade;
+-----+-----+-----+-----+
| plade_id | pladetitel | kunstnernavn | udgivelsesaar |
+-----+-----+-----+-----+
| 1 | Pladetitel 1 | Kunstner 1 | 1999 |
| 2 | Pladetitel 2 | Kunstner 2 | 2000 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from sangtitler;
+-----+-----+-----+-----+
| plade_id | sang_id | sangtitel |
+-----+-----+-----+-----+
| 1 | 1 | Sangtitel 1 |
| 1 | 2 | Sangtitel 2 |
| 2 | 3 | Sangtitel 3 |
| 2 | 4 | Sangtitel 4 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Transaktioner på pladekartoteket, fortsat

- Sletning af en plade fra kartoteket.

```
delete from plade where plade_id = 1;
```

```
mysql> select * from plade;
+-----+-----+-----+-----+
| plade_id | pladetitel | kunstnernavn | udgivelsesaar |
+-----+-----+-----+-----+
| 2 | Pladetitel 2 | Kunstner 2 | 2000 |
| 3 | Abbey Road | Beatles | 1969 |
+-----+-----+-----+-----+

mysql> select * from sangtitler;
+-----+-----+-----+-----+
| plade_id | sang_id | sangtitel |
+-----+-----+-----+-----+
| 1 | 1 | Sangtitel 1 |
| 1 | 2 | Sangtitel 2 |
+-----+-----+-----+-----+
```

Bemærk, at der ikke checkes for om der er sange tilknyttet den plade der slettes.

Transaktioner på pladekartoteket (transaktioner .mysql)

- Oprettelse af en ny plade i kartoteket (uden tilknyttede sange)
insert into plade (pladetitel, kunstnernavn, udgivelsesaar)
values ('Abbey Road', 'Beatles', 1969);
- Tilknytning af en ny sang til en eksisterende plade
insert into sangtitler (plade_id, sangtitel)
values (last_insert_id(), 'Here comes the sun');
- Sletning af en bestemt sang fra en plade
delete from sangtitler
where plade_id = 3
and sangtitel = 'Here comes the sun';
- Sletning af alle sange fra en plade
delete from sangtitler where plade_id = 2;

Transaktioner på pladekartoteket, fortsat

- Visning af sange på en plade med et givet plade_id (select)
select pladetitel, kunstnernavn, sangtitel
from plade, sangtitler
where plade.plade_id = sangtitler.plade_id
and plade.plade_id = '1';
- Visning af alle plader udgivet af en bestemt kunstner

```
select pladetitel, kunstnernavn, sangtitel  
from plade, sangtitler  
where plade.plade_id = sangtitler.plade_id  
and plade.kunstnernavn = 'Kunstner 1';
```

PHP3 (hello_world.php3)

I AOL-server kan vi enten skrive Tcl-programmer der genererer HTML kode eller adp-sider der er HTML kode med indlejret Tcl:

I PHP3 skriver man som regel HTML-kode med indlejret PHP3 kode:

```
<html>
<body>
<?php
$myvar = "Hello World";
echo $myvar;
?>
</body>
</html>
```

<?php indikerer starten af en blok med PHP3 kode.

?> afslutter en blok med PHP3 kode.

Alle variable, f.eks. myvar starter med et dollar tegn (\$)

echo udskriver indholdet af myvar.

Funktioner i PHP3 (Layout.php3)

Vi kan definere procedurerer (funktioner) så samme måde som i Tcl:

```
<?php
function html_pg ($title, $body) {
return "<html>
<head>
<title>". $title. "</title>
</head>
<body>$body</body>
</html>";
}
echo html_pg("Hello World", "Hello World");
?>
```

Bemærk igen \$ foran variabelnavne – også i parameterlisten.

echo udskriver (returnerer) resultatet af at kalde funktionen html_pg.

Strengsammensætning er ligeså simpelt som i Tcl – vi anvender punkum (.) til at sætte strenge sammen. I Tcl anvender vi append.

Indholdet af variable kan indsættes i en streng ved blot at skrive variabelnavn med foransatlet dollar tegn – præcis som i Tcl.

Bemærk, at dette script indholder kun PHP3 kode – præcis som vores Tcl-programmer kun indeholder Tcl kode.

Vi kan lave løkker som i Tcl (mult_tabel.php3)

```
<?php
function html_pg ($title, $body) { ... }

function mult_tabel($n) {
    $res = "<table border=1><tr><th>A</th><th>B</th><th>A times B</th></tr>\n";
    for ($x=0; $x<10; $x=$x+1) {
        $res = $res. "<tr><td>$x</td><td>$n</td><td>". $x*$n. "</td></tr>\n";
    }
    $res = $res. "</table>";
    return $res;
}
echo html_pg("Multiplikationstabel (3-tabel)", mult_tabel(3));
?>
```

De mest almindelige løkke- og betingelses-konstruktioner findes i PHP3:

- if-then-else konstruktionen.
- while-løkker
- switch konstruktionen

Formvariable i PHP3 (mult_tabel2.php3)

Formvariable er tilstede i programmet uden at vi behøver at gøre noget.

Programmet forventer formvariablen n – scriptet henviser blot til n.

```
<?php
function html_pg ($title, $body) { ... }
function mult_tabel($n) { ... }

if (ereg("^[0-9]+$", $n)) {
    echo html_pg("Multiplikationstabel ($n-tabel)", mult_tabel($n));
} else {
    echo html_pg("Multiplikationstabel -- fejl",
        "Du skal angive formvariablen n");
}
?>
```

Vi kan checke variablen n ved hjælp af regulære udtryk (ereg) – præcis som i Tcl.

Opgave: Hvad matcher det regulære mønster ovenfor?

PHP3 og MySQL

For at tilgå en MySQL database fra PHP3, så skal vi

- Anvende `mysql_connect` for at gå en forbindelse til databasen (håndtag)
- Anvende `mysql_select_db` for at vælge en database, f.eks. `dwp`.
- Anvende `mysql_query` til at sende forespørgslen til MySQL.
- Hente resultatet med `mysql_fetch_row`.
- Lukke forbindelsen med `mysql_close`

Eksempel der viser alle sange i kartoteket (`vis_sange_demo.php3`)

```
<?php
function html_pg ($title, $body) { ... }

$server = "mysql.itu.dk";
$user = "nh";
$password = "nh";

function error($msg) {
    print( "<h2>ERROR: $msg</h2>\n" );
    exit();
}
```

PHP3 og MySQL, fortsat (`vis_sange_demo.php3`)

```
/* Make a connection to the database server: */
$db = mysql_connect($server,$user,$password);

if (!$db)
    error( "Cannot open connection to $user@$server" );

/* Choose the database to work with: */
if (!mysql_select_db( "dwp", $db ))
    error( "Cannot select database 'dwp'. " );

$body = "Successfully opened connection to MySQL server $user@$server<p>\n";

$query = "select pladetitel, kunstnernavn, sangtitler
from plade, sangtitler
where plade.plade_id = sangtitler.plade_id";

$result = mysql_query($query,$db);
```

Vi åbner en forbindelse til databasen med bruger `nh` på maskine `mysql.itu.dk`.

Vi vælger databasen `dwp`.

Vi udfører en forespørgsel med `mysql_query`.

PHP3 og MySQL, fortsat (`vis_sange_demo.php3`)

```
$body = $body . "<table><tr><th>Pladetitel</th>
<th>Kunstnernavn</th><th>Sangtitel</th></tr>\n";
while ($row = mysql_fetch_row( $result )) {
    list($pladetitel, $kunstnernavn, $sangtitel) = $row;
    $body = $body . "<tr><td>$pladetitel</td><td>$kunstnernavn</td>
<td>$sangtitel</td></tr>";
}
$body = $body . "</table><p>\n";
echo(html_pg("Records", $body));

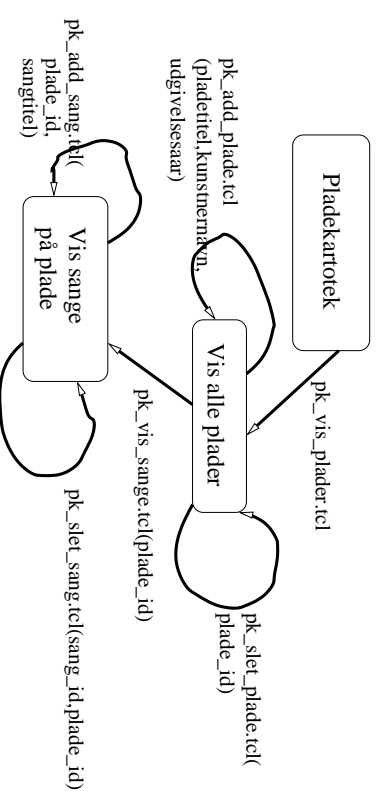
/* Close database: */
mysql_close($db);
?>
```

Rækkerne i resultatet hentes en efter en med `mysql_fetch_row`.

Row indeholder alle feltene for hver række, og vi anvender *pattern matching* når vi overfører resultaterne til variablene `pladetitel`, `kunstnernavn` og `sangtitel`.

Vi lukker databasen med `mysql_close`.

Pladekartoteket



Kasserne repræsenterer HTML kode, som brugeren ser.

Pilene repræsenterer PHP-programmer som f.eks. opdaterer databasen og genererer friske HTML sider til brugeren

Pladekartoteket, kode

Koden til de 6 filer findes online:

- `pk_vis_plader.php3:`
`http://www.it-c.dk/~nh/DWEB/Pladekartotek/pk_vis_plader.php3.txt`
- `pk_add_plade.php3:`
`http://www.it-c.dk/~nh/DWEB/Pladekartotek/pk_add_plade.php3.txt`
- `pk_slet_plade.php3:`
`http://www.it-c.dk/~nh/DWEB/Pladekartotek/pk_slet_plade.php3.txt`
- `pk_vis_sange.php3:`
`http://www.it-c.dk/~nh/DWEB/Pladekartotek/pk_vis_sange.php3.txt`
- `pk_add_sang.tcl:`
`http://www.it-c.dk/~nh/DWEB/Pladekartotek/pk_add_sang.php3.txt`
- `pk_slet_sang.tcl:`
`http://www.it-c.dk/~nh/DWEB/Pladekartotek/pk_slet_sang.php3.txt`

Problemer med Cookies

Der er desværre en del problemer med cookies:

- Almindelige browsere understøtter kun omkring 20 cookies pr. site
- Almindelige browsere understøtter kun omkring 300 cookies totalt
- En cookie kan højst være 4 kilobytes stor
- Sikkerhedsproblemer med Cookies

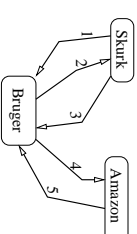
Hvis et site benytter cookies til identification af en person kan personen blive udsat for et "rekanis-attack".

Eksempel:

– En skurk sætter en webside op der omdirigerer (redirects) personer til Amazon's bestilingside for en bestemt bog (udenom alle forns)

– Herefter sender skurken en email til en "uskyldig person" som skurken ved køber ind på Amazon. skurken beder den "uskyldige person" om at "hænge grisen" (trykke på et link)

– Når den uskyldige person fanger grisen bliver der bestilt en bog uden at personen kan afværge købet - Amazon får tilsendt en cookie og accepterer bestillingen!



1. Email til brugger om at han skal "hænge grisen"
2. Brugger "fanger grisen", dvs aktiverer link hos skurken
3. Skurken sender en redirect til Brugger om at aktivere link hos Amazon, som egentlig er at bestille en bog
4. Brugger bestiller bog hos Amazon
5. I det brugger også sender cookie til Amazon, så bestilles bogen helt uden at brugeren skal afgive flere informationer.

Hvad er en Cookie

En cookie er en tekst-streng som en web-server sender til en browser og som browseren returnerer uændret når browseren igen besøger det samme site eller domain.

Cookies er nyttige til:

- Håndtering af simpel form for tilstand - session tracking
- Brugspecialisering (Customization)
- Fokusering af banner ads - hvad var man interesseret i sidst?
- Adgangskontrol – login mekanisme

I Netscape på Linux kan man se hvilke cookies der er installeret i filen `.netscape/cookies:`

```
# Netscape HTTP Cookie File
# http://www.netscape.com/newsref/std/cookie_spec.html
# This is a generated file! Do not edit.
linuxlab.dk FALSE / FALSE 1262307600 ad_browser_id 13
linuxlab.dk FALSE / FALSE 1262307600 last_visit 988628269
linuxlab.dk FALSE / FALSE 1262307600 second_to_last_visit 988628263
linuxlab.dk FALSE / FALSE 1262307600 ad_user_login 60.4C6F674E6432373032
```

Filformatet er som følger:

Domain	?	Path	Secure	Expires	Name	Value
linuxlab.dk	FALSE	/	FALSE	1262307600	ad_browser_id	13

Cookies som en Trussel mod Privatlivet

Der er en grund til at nogle personer vælger at slå cookies fra i deres browser:

- Søgmaskiner viser ads for hvad man søgte efter sidste gang – kan f.eks. være problematisk hvis man bliver kigget over skulderen af arbejdsgiveren.

Adgangskontrol i AOLserver

- Når et underkatalog forespørges, så kan AOLserver sættes til at afvikle et Tcl-script før siden returneres til brugeren.

- Tcl-scriptet kan kigge på en cookie – og dermed finde et kodeord gemt i cookien – og dermed checke at brugeren har adgang til servicen.

- Sikkerhedsrisici - er dette en sikker strategi?

- Se `http://www.photo.net/wtr/thebook/: Philipp and Alex's Guide to Web Publishing`, kapitel 16 (side 518-521) omkring authentication; og kapitel 15 (side 470) om hvorledes man sender en cookie til en browser:

```
# return a redirect with a cookie!
ns_write "HTTP/1.0 302 Found
Location: http://hug.it.edu:8077/Login/index.tcl
MIME-Version: 1.0
```

```
Set-Cookie: lg_user_id=expired;path=/Login; expires=Fri, 01-Jan-1990 01:00:00 GMT
Set-Cookie: lg_user_id=$user_id; path=/Login;
```

You should not be seeing this! "

Loginmekanisme – Brugertabel (Login.sql)

Vi har brug for en brugertabel med login og password:

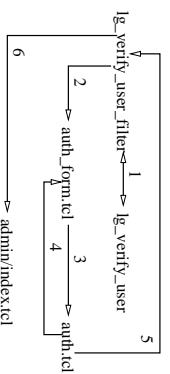
```
create table lg_user (
  user_id int primary key,
  password varchar(100) not null,
  login varchar(20) unique not null,
  name varchar(100) not null
);
```

USER_ID	PASSWORD	LOGIN	NAME
1	Niels	nh	Niels Hallenberg
2	Lise	llise	Lise Bennedsen

Loginmekanisme – Filerne

- /web/nh/tcl/ig_verify_user.tcl: Filter som anvendes hver gang man besøger en login-beskyttet side
- /web/nh/www/Login/auth_form.tcl: Den form, hvor man indtaster sit login og password
- /web/nh/www/Login/auth.tcl: Den checker om man findes i db og i så fald sætter en cookie med login og password samt redirect til admin/index.tcl.
- /web/nh/www/Login/logout.tcl: Den fjerner cookies, således at man præsenteres for en login side.
- /web/nh/www/Login/admin/index.tcl: Login-beskyttet side
- /web/nh/www/Login/user/index.tcl: Ikke login-beskyttet side.

Loginmekanisme – Overblik



- 1: cookie checkes i `lg_verify_user`
- 2: hvis login er forkert så returneres et loginbillede
- 3: indtastet login og password checkes ved at `auth.tcl` sætter en cookie med indtastet login og password, hvorefter der redirectes til `adminv/index.tcl`, dvs. `lg_verify_user` filer afvikles igen (5).
- 4: hvis login ikke findes i db, så sætter vi ikke en cookie men returnerer til loginbillede
- 6: login og password i cookie er godkendt og siden vises.

`lg_verify_user.filter`:

- udkøres for alle filer der hentes i `admin/`.
- kalder `lg_verify_user`
 - hvis `user_id = 0`, så returneres `auth_form.tcl`.
 - hvis `user_id ≠ 0`, så returneres fore-spurgte side.

`lg_verify_user`:

- finder login og password i cookies
- checker password og login med database
 - returnerer `user_id` hvis login er ok
 - returnerer 0 hvis login ikke er ok

Øvelse 11

Øvelse 11 er åben, dvs hvis du har en ide til en lille web-service som du godt vil bygge, så har du mulighed herfor.

Du kan hente inspiration på løbesejdel 11.

Du kan også vælge at regne prøveeksamen fra efteråret 2000:

<http://www.itu.dk/courses/W2/E2000/exm0.html>

Hvis du regner prøveeksamen, så skal du stadig aflevere elektronisk, dvs. du skal få servicen til at køre.

Vigtige datoer

- Udfærdigelse af projektrapport, tirsdag den 27/11-2002 klokken 10.00 - 12.00 (lokale oplyses senere). Forelæsning ilm. projekter i Grundlæggende Programmering. Læs <http://www.itu.dk/courses/GP/F2001/rapport.pdf>.
- Eksamen er den 15. januar 2002 – nærmere tidspunkt og sted annonceres når jeg får det oplyst.
- Eksamenssæt for E2000 diskuteres 8. januar klokken 10 – 12. Lokale oplyses senere
- Eksamenssæt for F2001 diskuteres 10. januar klokken 10 – 12. Lokale oplyses senere
- Spørgetime den 11. juni klokken 10 – 12. Lokale oplyses senere.