

Eksamen, DSDS, efterår 2010

Introduktion til Scripting, Databaser og Systemarkitektur

Javier San Agustin & Henrik Skovsgaard

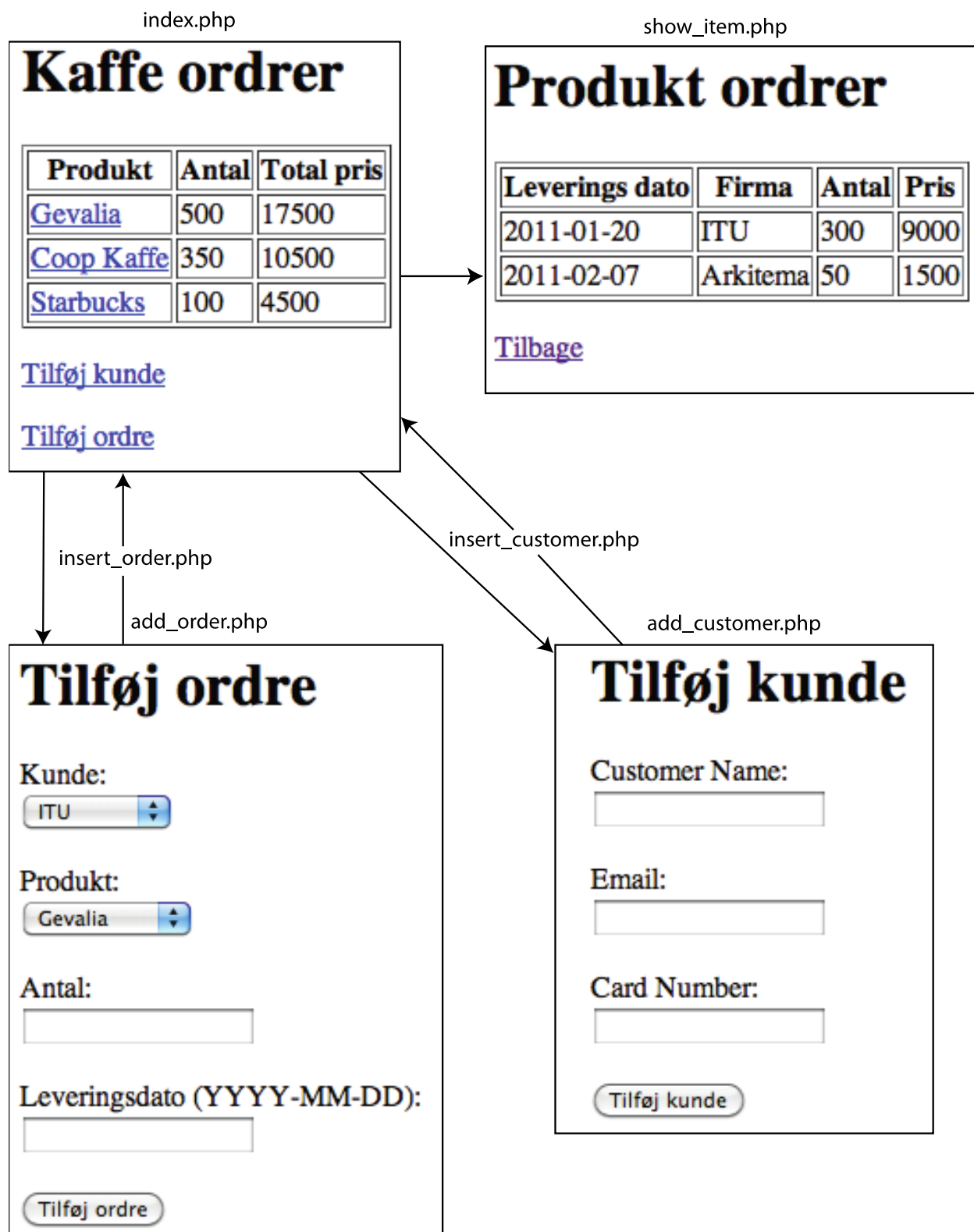
IT Universitetet i København

13. januar 2011

- Alle hjælpemidler er tilladte, dog ikke computer og kommunikationsmidler.
- Eksamen er skriftlig, fire timer (9-13), og den bedømmes fra 0 til 100 procent.
- Eksamen består af fem hovedopgaver, der alle ønskes løst.
- Skriv med kuglepen og kun på den ene side af papiret.
- Et godt råd: Gennemlæs opgavesættet inden du begynder at besvare de enkelte opgaver.
- Du bestemmer selv, om du benytter HTML eller XHTML, bare vær konsekvent.

I dette eksamenssæt skal du implementere dele af et kaffefirma kaldet Coffee'R'us. Vores firma er en lille virksomhed, der tager imod ordre fra en række kunder. Kunderne er virksomheder, der typisk bestiller et større parti kaffe.

Coffee'R'us kører som et internt system, hvor alle sider administreres af de ansatte (Figur 1). Systemet har en login-funktion, som du ikke skal implementere.



Figur 1 Site-map for det web-baserede system. Et screenshot (kasser) i diagrammet angiver de sider brugeren kan se i systemet. Hver sides navn er angivet ovenfor hvert billede. En pil i diagrammet indikerer, at der er et link, der kan bringe brugeren fra en side til en anden. En pil med tekst indikerer, at der bliver kørt et php-script før brugeren sendes videre mellem to sider.

Baggrund:

Filen `index.php` er systemets forside, og viser et overblik over ordrer fordelt på produkter. For hver produkt er mængden og prisen angivet. For hvert produkt er der et link til `show_item.php`. Linket sender produkt ID (`productid`) med til `show_item.php`.

Filen `show_item.php` lister følgende: den ønskede leveringsdato (`purchases.deliverydate`), firmanavn (`customers.name`), mængde (`purchases.amount`) samt pris (`purchases.totalprice`).

Filen `add_customer.php` indeholder en form, der kan tilføje en kunde til systemet.

Scriptet `insert_customer.php` modtager de forskellige inputs, validerer dem og indsætter data i databasen. Brugeren bliver derefter sendt til `index.php`.

Filen `add_order.php` indeholder en form, der kan tilføje en ordre til vores system.

Scriptet `insert_order.php` modtager de forskellige inputs, validerer dem og indsætter data i databasen. Brugeren bliver derefter sendt til `index.php`.

helpers.php:

I det følgende antages det, at funktionerne `error`, `mydb_connect`, `show_header`, `show_footer`, `check_text`, `check_int`, `check_email` og `check_date`, er tilgængelige i filen `helpers.php`.

Alle funktioner, du bliver bedt om at lave i dette opgavesæt, vil ligeledes være tilgængelige i filen til efterfølgende opgaver.

`error`: tager en streng som argument. Strengen udskrives og funktionen kalder `exit`, der stopper programmet.

`mydb_connect`: tager ingen argumenter og opretter forbindelse til systemets database.

`check_int`: tager et tal som argument og tjekker, om det er et gyldigt heltal. Er det ikke, kaldes funktionen `error` og programmet stopper.

`check_text`: tager en streng som argument og tjekker, om det indeholder ulovlige tegn. Indeholder strengen det, kaldes funktionen `error`, og programmet stopper.

`check_email`: tager en streng som argument og tjekker, om det er en lovlig email. Indeholder strengen en ulovlig mail, kaldes funktionen `error`, og programmet stopper.

`check_date`: tager en streng som argument og tjekker, om det er en lovlig dato i formatet: 'YYYY-MM-DD'. Indeholder strengen en ulovlig dato, kaldes funktionen `error`, og programmet stopper.

Koden for `show_header` og `show_footer`:

```
function show_header($title){  
    echo "<!DOCTYPE ....
```

```

<html><head>

<title>$title</title>

</head><body>

<h1>$title</h1>";

}

function show_footer(){

    echo "</body></html>";

}

```

Opgave 1 (10 point) – HTML

Opgave 1.1 (5 point) Opstil HTML-koden for tabellen der vises i index.php i Figur 1. Besvarelsen skal have formen:

```
<table> ... </table>
```

Teksten i den første række skal være i et <th> element.

Links i Produkter-kolonnen peger på show_item.php, og viser ordrene for de enkelte produkter. Det sender et productid med i URL'en i formen "show_item.php?productid=___".

- Gevalia har productid = 1
- Coop Kaffe har productid = 2
- Starbucks har productid = 3

Opgave 1.2 (5 point) Skriv HTML-koden for formen vist i add_customer.php i Figur 1. Besvarelsen skal have formen:

```
<form> ... </form>
```

Layoutet i formen er ikke relevant, det samme gælder beskrivende tekst.

Der er følgende krav til formen:

- Formens action attribut skal have værdien insert_customer.php og metoden post
- Formen indeholder følgende formvariable:
 - Et text felt med name attributten sat til name
 - Et text felt med name attributten sat til email
 - Et text felt med name attributten sat til card
 - Et submit felt med value attributten sat til Tilføj Kunde

Opgave 2 (5 point) – PHP

Opgave 2.1 (5 point) Siden `add_order.php` har to drop-down lister. Den ene liste indeholder kundenavne, og den anden indeholder produktnavne.

Du skal forestille dig, at du allerede har lavet en forespørgsel til databasen og resultatet er indlæst i variabelen `$rows`, der er en tabel i 3 rækker:

<code>products.productid</code>	<code>products.name</code>
1	Gevalia
2	Coop Kaffe
3	Starbucks

Du skal skrive en funktion, der kan generere HTML-koden for drop-down listen med produkter. Hver option-element i drop-down listen skal vise produktnavnet og have en value der korresponderer med productid. Resultatet skal se ud som i Figur 2.



Figure 2 Drop-down liste med produkter

Opgave 3 (10 point) – Regulære Udtryk

Opgave 3.1 (5 point) Skriv et regulært udtryk der tjekker om en sekvens af numre stemmer overens med et konto nummer, for eksempel: 9828-2424-1110-8354 og 3256-1212-9814-5623 men IKKE a828-2DD4-1330-12cc

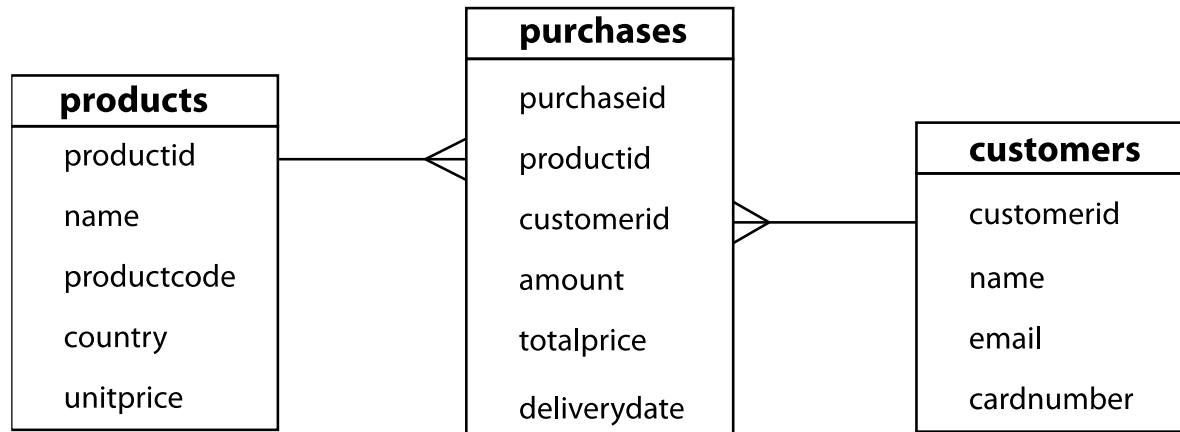
Opgave 3.2 (5 point) Skriv et regulært udtryk der undersøger om vores kunder kommer fra en Danmark eller en organisation. Derfor vil vi kun have e-mail adresser, der ender med `.dk` eller `.org`.

For eksempel: `javier@itu.dk`, `henrik@who.org` men IKKE `john@hotmail.com`.

Opgave 4 (25 point) – SQL/Datamodel

Baggrund:

Datamodellen er angivet nedenfor som et ER-diagram (Figur 3):



Figur 3 ER Diagram. Den øverste kasse angiver tabellens navn og indholdet angiver attributterne, der et tilknyttet de enkelte tabeller. En-til-mange relationer er angivet med "kragefødder".

Tabellerne i Figur 3 er lavet med følgende SQL-kommandoer:

```
CREATE TABLE products (  
    productid INT PRIMARY KEY AUTO_INCREMENT,  
    name VARCHAR(255) NOT NULL,  
    productcode VARCHAR(255) NOT NULL,  
    country VARCHAR(255) NOT NULL,  
    unitprice INT  
) ENGINE=InnoDB;  
  
CREATE TABLE customers (  
    customerid INT PRIMARY KEY AUTO_INCREMENT,  
    name VARCHAR(255) NOT NULL,  
    email VARCHAR(255) NOT NULL UNIQUE,  
    cardnumber VARCHAR(255) NOT NULL  
) ENGINE=InnoDB;  
  
CREATE TABLE purchases (  
    purchasesid INT PRIMARY KEY AUTO_INCREMENT,  
    productid INT,  
    customerid INT,  
    amount INT,  
    totalprice INT,  
    deliverydate DATE,  
    FOREIGN KEY (productid) REFERENCES products(productid),
```

```
FOREIGN KEY (customerid) REFERENCES customers(customerid)
) ENGINE=InnoDB;
```

```
INSERT INTO products VALUES ('', 'Gevalia', 'GV', 'Sweden', 35);
INSERT INTO products VALUES ('', 'Coop Kaffe', 'CK', 'Norway', 30);
INSERT INTO products VALUES ('', 'Starbucks', 'SB', 'USA', 45);
```

```
INSERT INTO customers VALUES ('', 'ITU', 'coffee@itu.dk', '4526-2323-4545-4545');
INSERT INTO customers VALUES ('', 'Nordea', 'per@nordea.dk', '8989-1212-2323-4252');
INSERT INTO customers VALUES ('', 'Arkitema', 'kaffe@baddesign.dk', '9828-2424-1110-8354');
```

```
INSERT INTO purchases VALUES('', 2, 1, 300, 9000, '2011-01-20');
INSERT INTO purchases VALUES('', 3, 3, 100, 4500, '2011-01-23');
INSERT INTO purchases VALUES('', 1, 2, 500, 17500, '2011-02-03');
INSERT INTO purchases VALUES('', 2, 3, 50, 1500, '2011-02-07');
```

Opgave 4.1 (5 point) Skriv en SQL kommando der henter kundenavne (customers.name) og antallet af ordrer tilknyttet den enkelte kunde. Outputtet af SELECT kommandoen skal resultere i følgende tabel:

customers.name	number of orders
ITU	1
Nordea	1
Arkitema	2

Hint:

- Du skal anvende COUNT og GROUP BY i din SQL kommando.

Opgave 4.2 (5 point) Indsæt en ny kunde i databasen med INSERT kommandoen. Kunden vil have følgende information:

- name: KU
- email: kaffe@ku.dk
- cardnumber: 1111-2222-3333-4444

Opgave 4.3 (5 point) Indsæt et nyt køb i databasen med den nye kunde (customerid = 4), der opretter en bestilling på 100 Starbucks kaffe til levering den 2011-02-20.

Opgave 4.4 (5 point) Skriv en SQL kommando der henter de værdier fra databasen, der kræves til at bygge tabellen i index.php. De ønskede værdier er:

- productid
- product name
- Samlede antal solgte enheder for hvert produkt
- Samlede pris for hvert produkt

Hints:

- Du skal anvende SUM og GROUP BY.

Opgave 4.5 (5 point) Hvilket tiltag er der gjort for at gøre de enkelte kunder unikke?

Hint: hvordan er tabellen oprettet?

Opgave 5 (30 point) – Webservice

Hint: Husk at inkludere helpers.php, hvis du ønsker at anvende funktionerne i filen.

Opgave 5.1 (7.5 point) Skriv scriptet til index.php. Scriptet tager ingen argumenter. Scriptet anvender databaseforespørgslen fra opgave 4.4 og bygger en HTML-tabel som vist i Figur 1. Første række i tabellen er header row og skal vises i fed skrifttype (bold) med et <th> element. De kommende rækker skal genereres automatisk med php.

Opgave 5.2 (7.5 point) Skriv scriptet til show_item.php. Scriptet tager et argument i URL'en (productid) og viser alle køb for et givent produkt. Scriptet skal generere en HTML tabel, der viser Leverings dato (purchases.deliverydate), Firma (customers.name), Antal (purchases.amount) og total Pris (purchases.totalprice) for ordren, som vist i Figur 4. Tabellen skal designes på samme vis som i opgave 5.1.

Produkt ordrer			
Leverings dato	Firma	Antal	Pris
2011-01-20	ITU	300	9000
2011-02-07	Arkitema	50	1500
Tilbage			

Figur 4 Produkt ordrer.

Opgave 5.3 (7.5 point) Skriv scriptet til `insert_customer.php`. Scriptet modtager data fra `add_customer.php` formen, validerer inputtet og indsætter værdierne i databasen. Dernæst sendes brugeren videre til `index.php`.

Opgave 5.4 (7.5 point) Skriv scriptet til `insert_order.php`. Scriptet modtager data fra `add_order.php` formen, validerer inputtet og indsætter værdierne i databasen. Dernæst sendes brugeren videre til `index.php`.

Hint: et af felterne i `purchases`-tabellen på databasen, kræver at `totalprice` skal beregnes. Denne pris findes ved at multiplicere antal (`amount`) med enhedsprisen (`unitprice`):

```
$totalprice = $amount * $unitprice;
```

Værdien for enhedsprisen (`unitprice`) er gemt i `products`-tabellen på databasen. Du kan hente denne værdi og gemme den i variabelen `$unitprice` med følgende kodeeksempel:

```
$rows = mysql_query("SELECT products.unitprice FROM products WHERE  
products.productid = $product");
```

```
$row = mysql_fetch_array($rows);
```

```
$unitprice = $row[0];
```

Opgave 6 (20 point) – Afrunding

Opgave 6.1 (5 point) Forklar kort og med egne ord hvorfor vi anvende løkker.

Opgave 6.2 (5 point) Forklar kort og med egne ord hvorfor vi anvender regulære udtryk.

Opgave 6.3 (5 point) Forklar kort og med egne ord hvad en funktion er, og hvorfor vi anvender dem.

Opgave 6.4 (5 point) Forklar kort med egne ord hvad de følgende teknologier anvendes til:

- HTML
- PHP
- SQL