

Eksamen, DSDS, efterår 2008

Introduktion til Scripting, Databaser og Systemarkitektur

Jonas Holbech
IT Universitetet i København*

6. januar 2009

- *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*

Introduktion

I dette eksamenssæt skal du implementere dele af det fiktive system ”www.lejlighed.nu”.

www.lejlighed.nu er en databasestyreret portal, hvor ejendomsmæglere kan sætte deres lejligheder til salg.

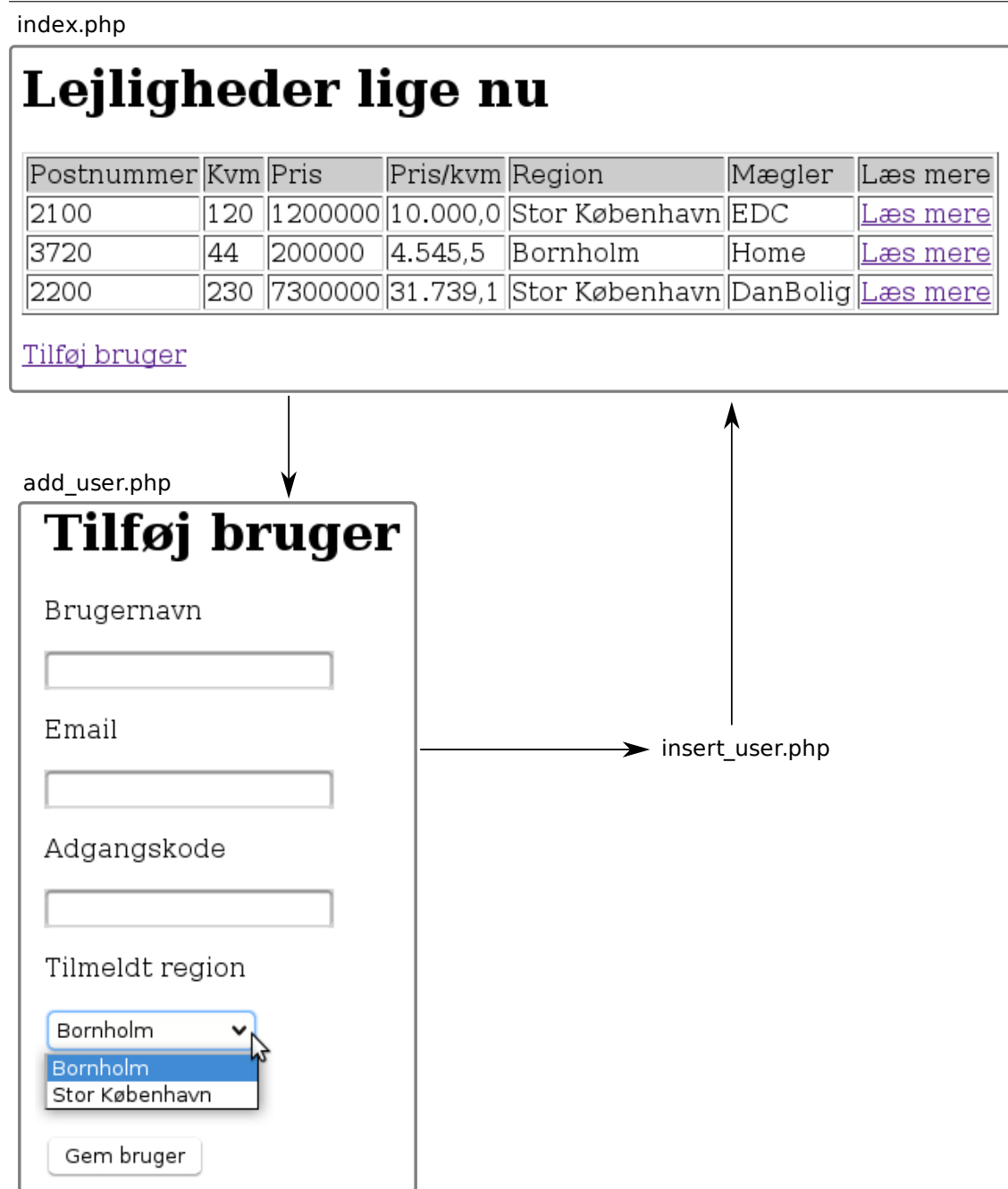
Brugere kan oprettes i systemet og få sendt en mail, når der kommer nye lejligheder i den region, de er tilknyttet.

Tanken er, at når en ejendomsmægler tilføjer en ny lejlighed i en region, eksempelvis Bornholm, modtager alle brugere, der er tilknyttet regionen, en mail.

Systemet er kun et udkast og er foreløbigt kun tiltænkt administratorere. Almidelige besøgende har således ikke adgang, og nye brugere kan kun oprettes af administratoren.

*Adresse: Rued Langgaards Vej 7, 2300 København S, Danmark. Email: holbech@itu.dk.

Figur 1 viser et site-map for det ønskede system.



Figur 1: Site-map for det web-baserede system. Et screenshot (kasser) i diagrammet angiver sider, brugeren kan se i systemet. Sidernes navne er angivet ovenfor hvert billede. En ikke-annoteret pil i diagrammet angiver, at en bruger kan klikke på et link i en side for at springe til en anden side. En annoteret pil i diagrammet angiver, at brugeren ved at klikke på en knap forårsager, at php-scriptet annoteret på pilen bliver kørt på serveren, og at php-scriptet videresender brugeren til en ny side.

Baggrund:

Filen `index.php` indeholder en liste over alle lejligheder i systemet. For hver lejlighed er der et link til `show_flat.php` (ikke med på Figur 1) og et link til `add_user.php`

Filen `add_user.php` indeholder en form, hvor administratoren kan oprette brugere, formens indhold sendes til `insert_user.php`

Filen `insert_user.php` tjekker indtastningerne og angiver fejlmeddelelser eller indsætter brugeren.

Bemærk: I det følgende antages det, at funktionerne `error`, `mydb_connect`, `html_header`, `html_footer`, `check_text`, `check_email` og `check_int` er tilgængelige i filen `utilities.php`. Alle funktioner, du bliver bedt om at lave i dette eksamenssæt, er ligeledes tilgængelige i filen.

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

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

`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 strengen indeholder skadelige dele. Gør den det, kaldes funktionen `error` og programmet stopper.

`check_email`: tager en streng som argument og tjekker om det er en gyldig email. Er det ikke kaldes funktionen `error` og programmet stopper.

Koden for `html_header` og `html_footer`:

```
function html_header($title){
    echo "<html><head>
        <title>$title</title>
    </head><body>
        <h1>$title</h1>";
}

function html_footer(){
    echo "</body></html>";
}
```

Opgave 1 (15 procent) - HTML

Opgave 1.1 (7 procent)

Din Opgave:

Lav HTML-koden for formen, svarende til `add_user.php` fra Figur 1. Besvarelsen skal have formen:

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

Bemærk: Layoutet i formen er ikke relevant, tekstuelle beskrivelser er ligeledes ikke nødvendige. (eks. 'Adgangskode:' eller 'Brugernavn')

Der er følgende krav til formen.

- Formens `action` attribut skal have værdien `insert_user.php`, og formens metode skal være `post`
- Formen indeholder fem formelementer
 - Et `text` felt med `name` attributten sat til `name`
 - Et `text` felt med `name` attributten sat til `email`
 - Et `password` felt med `name` attributten sat til `pass`
 - Et `select` felt med `name` attributten sat til `regionid`. Select elementet indeholder to `option`-elementer:
 - * Første element har `value` 1 og værdien `Bornholm` og skal være valgt
 - * Første element har `value` 2 og værdien `Stor København`
 - Et `submit` felt med `value` attributten sat til `Gem bruger`

Bemærk: På billedet i figur Figur 1 er `select` menuen foldet ud

Opgave 1.2 (8 procent)

Din Opgave:

Lav HTML-koden for tabellen, svarende til `index.php`

Bemærk: Hver række indeholder et "læs mere" link, der peger på `show_flat.php`. Linket sender et `flatid` med (a la `show_flat.php?flatid=2`)

Farver og lignende er ikke nødvendige

- Første række har `flatid` 2, anden række `flatid` 3 og tredje række `flatid` 1

Opgave 2 (20 procent) - PHP

Opgave 2.1 (7 procent)

Funktionen `sqm_price` regner kvadratmeterprisen ud på lejligheder, og returnerer tallet rundet af til en (1) decimal.

Funktionen tager to argumenter, antallet af kvadratmeter og lejlighedens pris.

Funktionen skal benytte funktionen `number_format` til at runde af og præsenterer tallet med danske decimaler. Dvs. ,(komma) som kommaseperator og .(punktum) som tusinde separator.

Din Opgave:

Skriv funktionen `sqm_price`

Hint: `number_format` tager fire argumenter, tallet der skal formateres, antallet af decimaler, kommaseperator og tusindeseperator

Opgave 2.2 (7 procent)

Baggrund:

Når systemet er færdigt skal brugerne kunne bedømme de enkelte lejligheder, og for at gøre rettelser lettere ønskes en funktion, der bygger en "rate" form.

Din Opgave:

Lav funktionen `rate_it`. Funktionen tager et argument, et id (`flatid`) på den lejlighed, der bliver bedømt.

Funktionen skal udskrive en form med fem radioknapper (med værdierne 1 til 5), til dette skal der benyttes en løkke. Formens `action` skal være `rate_it.php`, og `method` skal være `post`.

Formen skal indeholde et `hidden` felt med `name`-attributten sat til `flatid` og `value` sat til det argument, funktionen modtager. Radioknappernes `name`-attribut skal være `rating`

Formen skal derudover have en submit knap.

Opgave 2.3 (6 procent)

Din Opgave:

Skriv funktionen `send_mail`. Funktionen tager tre argumenter; Modtagerens email adresse, en overskrift (subject) og selve teksten i mailen (body).

Funktionen skal sende en mail med subject og body til modtageren. Mailen skal sendes fra mailadressen `noreply@lejlighed.nu`.

Hvis det lykkes at afsende en mail, skal der returneres `true`, ellers returneres `false`

Hint: Du får brug for PHP funktionen `mail`

For at sætte afsenderen skal du benytte `mail`-funktionens fjerde parameter.

Opgave 3 (10 procent) - Regulære Udtryk

Bemærk: Det forventes, at du benytter `^` og `$`, hvor det er relevant

Opgave 3.1 (5 procent)

Din Opgave:

Skriv et regulært udtryk, der matcher heltal, der er større eller lig med 50000 (halvtredstusinde)

Udtrykket skal således matche tallene 50000 og 89662, men ikke 50568.6 eller 56

Opgave 3.2 (5 procent)

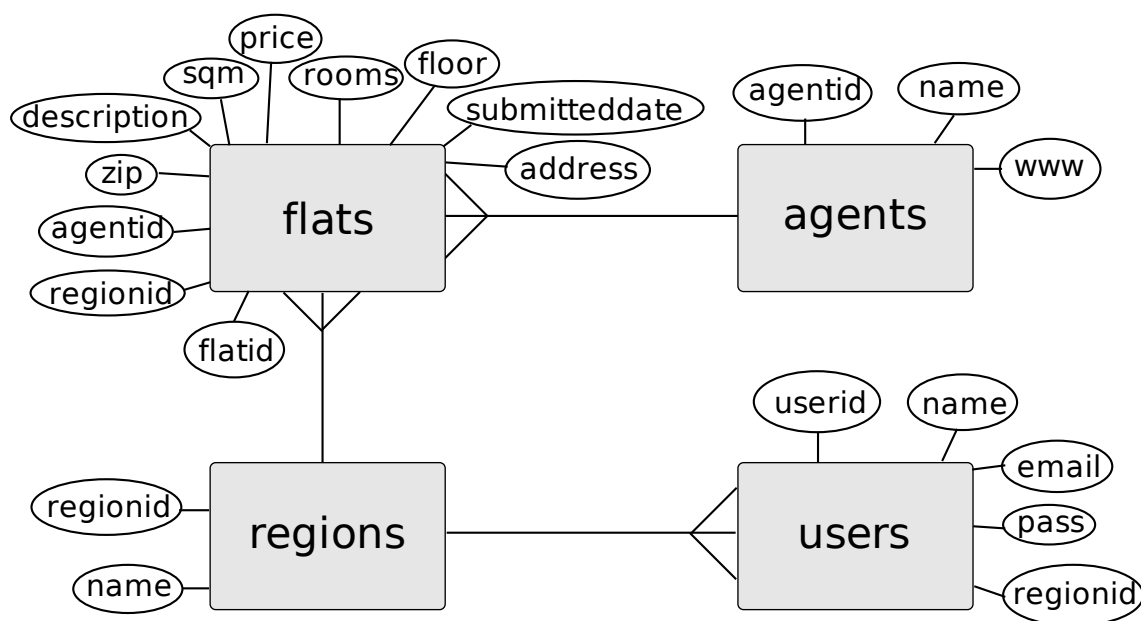
Din Opgave:

Skriv et regulært udtryk, der matcher danske postnumre, her begrænset til tal mellem 1000 og 9999

Opgave 4 (25 procent) - SQL / Datamodel

Baggrund:

Datamodellen for systemet er angivet i Figur 2 som et ER-diagram.



Figur 2: E/R-diagram for systemet. De firkantede kasser angiver entiteterne (tabellerne), og de ovale cirkler angiver attributter (felter), tilknyttet de enkelte tabeller. En-til-mange relationer er angivet med "kragefødder".

Tabellerne `regions`, `agents` og `users` er oprettet i MySQL med følgende SQL-kommandoer:

```
CREATE TABLE regions (
    regionid INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(250)
) ENGINE=InnoDB;
```

```
INSERT INTO regions VALUES
    ('', 'Bornholm'),
    ('', 'Stor København');
```

```
CREATE TABLE agents (
    agentid INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255),
    www VARCHAR(255) UNIQUE
) ENGINE=InnoDB;
```

```

INSERT INTO agents VALUES
  ('', 'Home', 'http://home.dk'),
  ('', 'DanBolig', 'http://danbolig.dk'),
  ('', 'EDC', 'http://edc.dk');

CREATE TABLE users (
  userid INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(250),
  email VARCHAR(250) UNIQUE,
  pass VARCHAR(40),
  regionid INT,
    INDEX(regionid),
  FOREIGN KEY (regionid) REFERENCES regions(regionid)
) ENGINE=InnoDB;

INSERT INTO users
  (name,email,pass,regionid)
VALUES
  ('Anders', 'anders@itu.dk', 'andersssh', 1),
  ('Betty', 'betty@itu.dk', 'bettyssh', 2),
  ('Carl', 'carl@itu.dk', 'carlssh', 1);

```

Opgave 4.1 (5 procent)

Konstruér en SQL-kommando til at oprette tabellen **flats**. Tabellen skal have følgende felter:

- **flatid** skal være af typen **INT**, være primær nøgle og benytte **AUTO_INCREMENT**
- **regionid** skal være af typen **INT** og referere til feltet **regionid** i tabellen **regions**
- **agentid** skal være af typen **INT** og referere til feltet **agentid** i tabellen **agents**
- **zip** skal være af typen **INT** og må højest være på 4 tegn
- **description** skal være af typen **TEXT**
- **sqm** skal være af typen **INT** og må højest være på 3 tegn
- **price** skal være af typen **INT**
- **rooms** skal være af typen **INT** og må højest være på 2 tegn
- **floor** skal være af typen **VARCHAR** og må højest være på 30 tegn
- **submitteddate** skal være af typen **DATE**
- **address** skal være af typen **TEXT**

Opgave 4.2 (5 procent)

Konstruér INSERT-kommandoer, der indsætter følgende relationer/lejligheder i tabellen **flats**:

- 4 værelser, 120 kvm, 2 sal tv, i storkøbenhavn, med postnummer 2100, adressen 'Østerbrogade 23', 1.200.000 kroner, sat til salg af EDC
- 1 værelse, 44 kvm, 3 sal th, på Bornholm, med postnummer 3720, adressen 'Vestermarievej 17', 200.000 kroner, sat til salg af Home
- 7 værelser, 230 kvm, 1 & 2 sal tv, i storkøbenhavn, med postnummer 2200, adressen 'Fensmarkgade 24', 7.300.000 kroner, sat til salg af DanBolig

Bemærk: Du bestemmer selv om du vil løse ovenstående som en eller flere kommandoer.

Du kan benytte MySQL's NOW() funktion til datoen.

Beskrivende tekst (feltet **description**) vælger du selv.

Opgave 4.3 (5 procent)

Din Opgave:

Skriv en SQL-kommando, der henter alle mæglere (**name**), der ikke har nogle lejligheder tilknyttet i tabellen **flats**.

Hint: Du kan blandt andet få brug for COUNT et LEFT JOIN og en HAVING betingelse. Du må gerne hente andre felter end **name**.

Opgave 4.4 (5 procent)

Din Opgave:

Skriv en SQL-kommando, der henter alle lejligheder (**flats.flatid**) samt navnet på mægleren der sælger lejligheden (**agents.name**). Der ønskes kun lejligheder der; har postnummeret 2200, koster under en halv million (500000) og har tre eller flere værelser.

Opgave 4.5 (5 procent)

Din Opgave:

Vis i tabelform outputet fra SQL-kommandoen i opgave 4.4

Opgave 5 (30 procent) - Web Service

Opgave 5.1 (7 procent)

Baggrund:

`index.php` er oversigten over alle lejligheder i systemet og er bygget ud fra følgende SQL-kommando:

```
SELECT flatid, zip, sqm, price, regions.name AS rname, agents.name AS aname
  FROM flats, agents, regions
 WHERE regions.regionid=flats.regionid
    AND agents.agentid=flats.agentid
 ORDER BY submitteddate DESC
```

Din Opgave:

Skriv hele scriptet `index.php`

Bemærk:

Scriptet modtager ingen formvariable.

Funktionerne `html_header` og `html_footer` skal benyttes.

Funktionen `sqm_price` skal benyttes til at udregne og formatere kvadratmeter prisen.

Du skal selv inkludere nødvendige filer.

Baggrundsfarver er ikke nødvendige.

I stedet for at skrive/kopiere hele SQL'en, kan du bare benytte "`SELECT . . . .`".

Opgave 5.2 (7 procent)

Din Opgave:

Skriv funktionen `delete_agent`. Funktionen tager ét argument. Et `agentid`.

Funktionen skal slette passende rækker i databasen ud fra `agentid`.

Hint:

Husk, `agentid` forekommer i to tabeller, og `agentid` i tabellen `flats` peger på (references) `agentid` i tabellen `agents`

Opgave 5.3 (8 procent)

Baggrund:

Scriptet `insert_user.php` modtager formvariabler fra `add_user.php`.

Hvis emailadressen ikke findes allerede, indsættes en række i tabellen `users`, og der sendes en mail til den nye bruger

Din Opgave:

Skriv scriptet `insert_user.php`

Hint:

Du skal selv inkludere nødvendige filer.

Forslag til fremgangsmåde

1. Tjekk formvariabler
2. Tjekk om mailen allerede findes, og kald funktionen `error`, hvis den gør
3. Indsæt en række i tabellen
4. Send en mail med adgangskoden ved hjælp af funktionen `send_mail` fra opgave 2.3
5. Send brugeren tilbage til `index.php` ved hjælp af funktionen `header`

Opgave 5.4 (8 procent)

Baggrund:

Den nuværende struktur tillader kun, at en bruger kan abonnere på én region af gangen. Feltet `email` i tabellen `users` er unik, så man kan kun tilmelde sig en gang, og referencen `regionid` i tabellen `users` kan kun pege på en region.

Din Opgave:

Kom med et eller flere forslag til, hvordan systemet kunne udvides, så man kunne implementere et smartere mail system.

Bemærk: Der ønskes en kort diskussion af løsningen, ikke en fuld implementering.

Opgaven må gerne besvares med et "ER-diagram"