

# Reeksamen, DSDS, forår 2008

Introduktion til Scripting, Databaser og Systemarkitektur

Jonas Holbech  
IT Universitetet i København\*

22. august 2008

- *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.*
- *Eksamen må besvares på dansk eller engelsk.*
- *Et godt råd: Gennemlæs opgavesættet inden du begynder at besvare de enkelte opgaver.*

## Introduktion

I dette eksamenssæt skal du implementere dele af det fiktive system "PublishIT".

PublishIT er et "Publishing Tool". Et system hvor brugerne i samarbejde kan skrive forskellige typer af artikler.

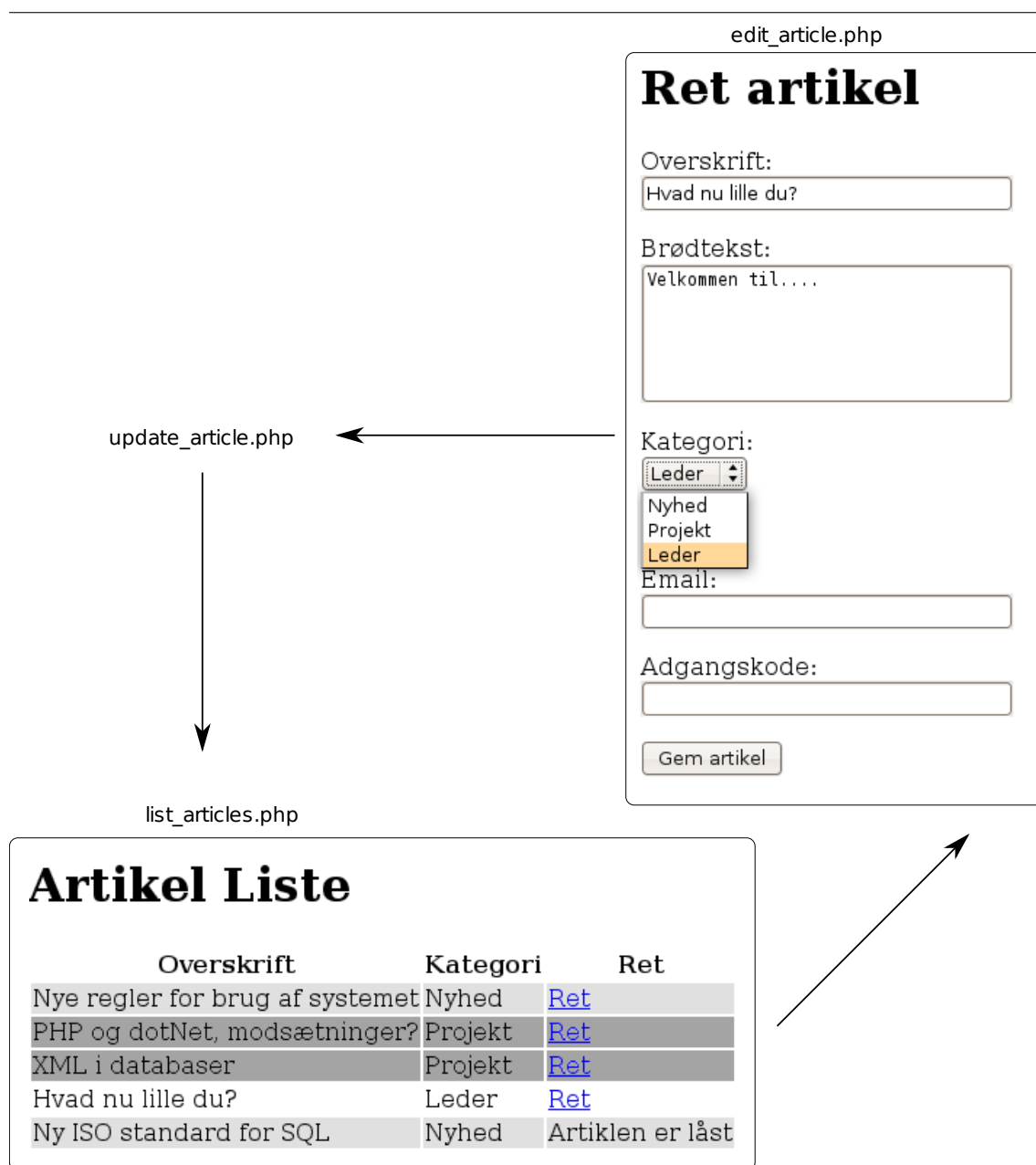
Eksamenssættet beskæftiger sig primært med redigering af eksisterende artikler.

Systemet tillader at en given artikel kan have mere end en forfatter og indeholder en simpel mekanisme til at tjekke om en given artikel allerede rettes af en anden bruger.

---

\*Adresse: Rued Langgaards Vej 7, 2300 København S, Danmark. Email: [holbech@itu.dk](mailto: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 navn 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 `list_articles.php` viser alle artikler i systemet, med farveangivelser for de forskellige artikel-kategorier. For hver artikel er artiklens overskrift (**header**) og kategori (**kategori**) angivet samt et link til redigering for de artikler der ikke allerede bliver rettet. Linket sender artiklens id (**aid**) med til `edit_article.php`

`edit_article.php` modtager en artikels id (**aid**) og indeholder en form hvor brugeren kan rette en eksisterende artikel. Formen er på forhånd udfyldt fra databasen. Brugeren skal indtaste email og adgangskode for at kunne gemme artiklen. Når siden besøges sættes **beingedited** til 1 (TRUE) i artiklens række i databasen, så flere brugere ikke kan rette samme artikel. Formens indhold sendes til `update_article.php`.

`update_article.php` tager det sendte indhold fra `edit_article.php`, validerer inputtet, tjekker om brugeren har rettigheder til at redigere artiklen og opdaterer databasen hvis alt er i orden. **beingedited** i tabellen **articles** sættes til 0 (**false**) så artiklen igen kan rettes. Herefter sendes brugeren tilbage til `list_articles.php`

**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($title){
    echo "<!DOCTYPE ....
    <html>
        <head>
            <title>$title</title>
        </head>
        <body>";
}
```

```
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 `edit_article.php` fra Figur 1. Besvarelsen skal have formen:

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

**Bemærk:** Layoutet i formen er ikke relevant, tekstuelle beskrivelser er ligeledes ikke nødvendige. (eks. 'Overskrift:' eller 'Ret artikel')

Indholdet af felterne (eks. 'Velkommen til...' og 'Leder' skal medtages. Der er følgende krav til formen.

- Formens `action` attribut skal have værdien `update_article.php` og formens metode skal være `post`
- Formen indeholder syv formelementer
  - Et `hidden` felt med `name` attributten sat til `aid` og `value` sat til `4`
  - Et `text` felt med `name` attributten sat til `header` og med `value` sat til 'Hvad nu lille du?'
  - Et `textarea` felt med `name` attributten sat til `body` og med indholdet sat til 'Velkommen til ...'
  - Et `select` felt med `name` attributten sat til `category`. Select elementet indeholder tre `option`-elementer:
    - \* Første element har værdien `Nyhed`
    - \* Andet element har værdien `Projekt`
    - \* Tredie element har værdien `Leder` og skal være valgt
  - Et `text` felt med `name` attributten sat til `email`
  - Et `password` felt med `name` attributten sat til `password`
  - Et `submit` felt med `value` attributten sat til `Gem Artikel`

**Bemærk:** På billedet i figur Figur 1 er `select` menuen foldet ud, det er kun for at vise indholdet.

## Opgave 1.2 (8 procent)

### *Din Opgave:*

Opstil HTML-koden for tabellen på siden `list_articles.php` i Figur 1. Besvarelsen skal have formen:

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

Den første række er overskrifterne på kolumnerne, og dem behøver du IKKE lave. Dit svar behøver altså kun have fem rækker.

Hver række i tabellen skal have en `class` attribut (css) med følgende værdier:

- Projekter (mørkegrå) skal have klassen (class) `'project'`
- Nyheder (lysee-grå) skal have klassen (class) `'news'`
- Ledere (hvide) skal have klassen (class) `'editorial'`

Links i "Ret" kolonnen peger på `edit_article.php` og sender et `aid` med i url'en.

- "Nye regler for brug af systemet" har `aid: 1`
- "PHP og dotNet, modsætninger?" har `aid: 2`
- "XML i databaser" har `aid: 3`
- "Hvad nu lille du?" har `aid: 4`
- "Ny ISO standard for SQL" har intet link, men teksten "Artiklen er låst"

**Hint:** `class` attributten kan du sætte på hver `<td>` eller `<tr>`

## Opgave 2 (20 procent) - PHP

### Opgave 2.1 (7 procent)

#### *Baggrund:*

Som nævnt tidligere får hver række i artikeloversigten en klasse ud fra artiklens kategori.

***Din Opgave:***

Skriv funktionen `set_class`.

Funktionen skal tage en streng som argument (en artikels kategori, eks Leder eller Projekt) og returnerer navnet på den tilsvarende klasse.

- har argumentet værdien 'Leder' skal strengen 'editorial' returneres
- har argumentet værdien 'Projekt' skal strengen 'project' returneres
- har argumentet værdien 'Nyhed' skal strengen 'news' returneres

**Opgave 2.2 (7 procent)*****Baggrund:***

Til brugere der har glemt deres adgangskode ønskes der en funktion der sender deres password via email.

***Din Opgave:***

Skriv funktionen `send_password`.

Funktionen tager et tal som argument svarende til feltet `aid` fra tabellen `authors`.

Funktionen skal hente `pass` (password) og `email` fra `authors` tabellen ud fra `aid` og sende en email indeholdende adgangskoden. Se evt Figur 2

**Hint:** Du kan antage at der allerede er oprettet forbindelse til databasen.

**Hint:** Ud over hvad du selv finder nødvendigt og de generelle mysql funktioner får du brug for funktionen `mail`

**Hint:** Teksten i emailen behøver kun at være adgangskoden og subject feltet kan du selv vælge.

**Opgave 2.3 (6 procent)*****Baggrund:***

Til nye brugere skal der kunne autogenereres en ny adgangskode. Til dette har vi funktionen nedenfor:

```
function generate_password($numbers) {
    $letters = array('a', 'b', 'c', 'd', 'e', 'f', 'g', 'h');

    for ( $i=0 ; $i < sizeof($numbers) ; $i++ ){
        $index=$numbers[$i];
        echo $letters[$index];
    }
}
```

Funktionen tager et array indeholdende tal som argument og udskriver en streng ud fra tallene i arrayet. Tanken er at funktionen `rand` er brugt til at lave de følgende arrays.

### ***Din Opgave:***

Hvad udskriver følgende to funktionskald:

```
$input = array(5,7,3,2,4);
generate_password($input);
```

```
$input = array(1,2,3);
generate_password($input);
```

## **Opgave 3 (10 procent) - Regulære Udtryk**

**Bemærk:** Når du laver dine regulære udtryk skal du benytte `$` og `^` hvor det er relevant.

### **Opgave 3.1 (5 procent)**

#### ***Baggrund:***

Ud over at tjekke tekst for SQL-injections og lignende vil vi gerne tjekke at indholdet i en artikel er af en vis længde. Her forestiller vi os et tal der angiver antallet af tegn i artiklen.

#### ***Din Opgave:***

Skriv et regulært udtryk der matcher gyldige heltal der er fra 100 og opefter (dvs. større end 99)

Det regulære udtryk skal således matche strengene `'987'`, `'100'` og `'6489'`, men ikke strengene `'0111'` eller `'99'`.

### Opgave 3.2 (5 procent)

***Baggrund:***

Når der oprettes brugere i systemet skal deres brugernavne tjekkes. For at brugerne ikke kalder sig eksempelvis administrator og kan snyde de øvrige brugere.

***Din Opgave:***

Skriv et regulært udtryk der matcher strenge der indeholder strengene 'admin' eller 'moderator'.

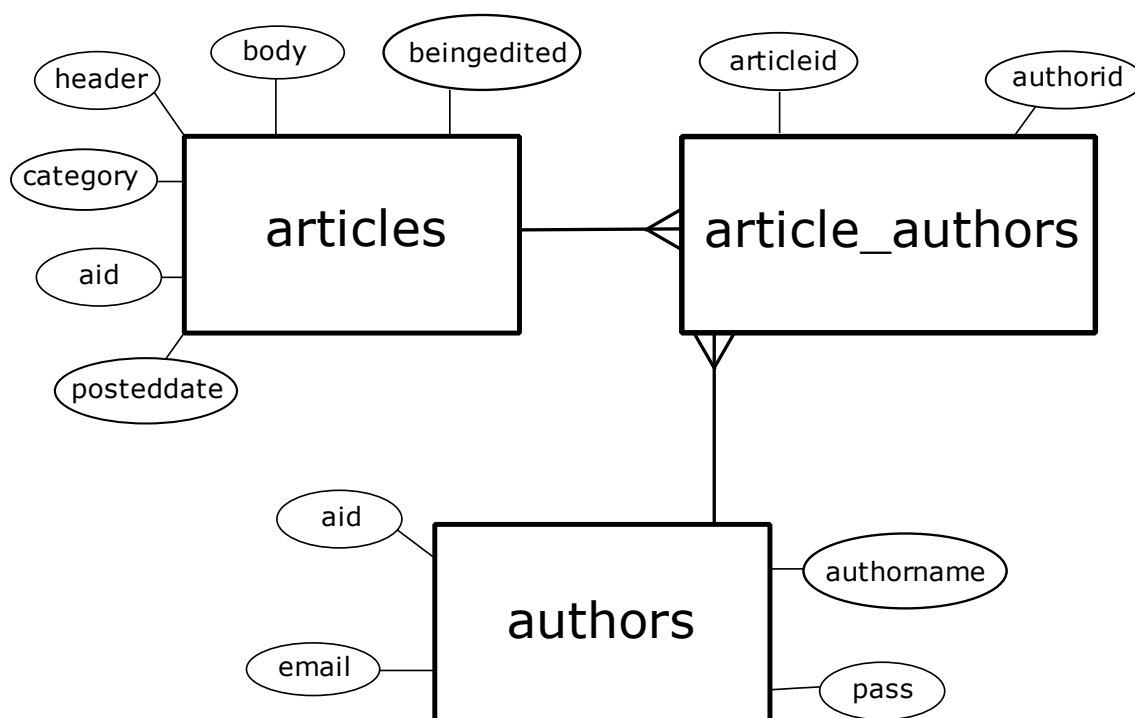
Det regulære udtryk skal således matche strengene 'Allan Admin Hansen', 'smoderators' og 'admine', men ikke strengene 'anders', 'Jens Hansen' eller 'mads\_find'.

**Bemærk:** Et af de to ord skal bare forekomme et sted i brugernavnet.

## 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 attributer (felter) tilknyttet de enkelte tabeller. En-til-mange relationer er angivet med "kragefødder".

---

Tabellerne `articles` og `authors` er oprettet i MySQL med følgende SQL-kommandoer:

```

CREATE TABLE articles (
  aid INT PRIMARY KEY AUTO_INCREMENT NOT NULL,
  posteddate DATETIME NOT NULL,
  category VARCHAR(200) NOT NULL,
  header VARCHAR(200) NOT NULL,
  body TEXT NOT NULL,
  beingedited INT(1) NOT NULL
) ENGINE=InnoDB;
  
```

```

INSERT INTO articles (aid, posteddate, category, header, body, beingedited) VALUES
('', NOW(), 'Nyhed', 'Nye regler for brug af systemet', 'Fra den 1/6 ....', 0),
  
```

```

('', NOW(), 'Projekt', 'PHP og dotNet, modsætninger?', 'Artiklen forsøger at...', 0),
('', NOW(), 'Projekt', 'XML i databaser', 'XML har længe været.....', 0),
('', NOW(), 'Leder', 'Hvad nu lille du?', 'Velkommen til....', 0),
('', NOW(), 'Nyhed', 'Ny ISO standard for SQL', 'MySQL rykker nu tættere på....', 1);

CREATE TABLE authors (
  aid INT PRIMARY KEY AUTO_INCREMENT NOT NULL,
  pass VARCHAR(40) NOT NULL,
  email VARCHAR(200) NOT NULL UNIQUE,
  authorname VARCHAR(200) NOT NULL
) ENGINE=InnoDB;

INSERT INTO authors (aid, pass, email, authorname) VALUES
('', 'sral', 'lars@itu.dk', 'Lars'),
('', 'allu', 'ulla@itu.dk', 'Ulla'),
('', 'tneb', 'bent@itu.dk', 'Bent'),
('', 'snah', 'hans@itu.dk', 'Hans'),
('', 'mik', 'kim@itu.dk', 'Kim'),
('', 'evot', 'tove@itu.dk', 'Tove');

```

#### Opgave 4.1 (5 procent)

Konstruér en SQL-kommando til at oprette tabellen `articleauthors`. Tabellen skal have følgende egenskaber:

- Feltet `articleid` skal være af typen `INT` og referere til `articles.aid`
- Feltet `authorid` skal være af typen `INT` og referere til `authors.aid`
- Kombinationen af `articleid` og `authorid` skal være unik, så en given "author" ikke kan være forfatter til en given artikel mere end en gang.
- Ingen af feltene må kunne indeholde værdien `NULL`

#### Opgave 4.2 (5 procent)

Konstruér `INSERT`-kommandoer der indsætter følgende relationer i tabellen `articleauthors`:

- Bent har skrevet artiklen "Hvad nu lille du?"
- Ulla har skrevet artiklen "XML i databaser"
- Lars har skrevet artiklen "Hvad nu lille du?"
- Bent har skrevet artiklen "Nye regler for brug af systemet"

- Kim har skrevet artiklen ”PHP og dotNet, modsætninger?”
- Kim har skrevet artiklen ”Ny ISO standard for SQL”

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

### Opgave 4.3 (5 procent)

Lav en SQL-kommando der henter `authorname`, forfatterens `aid` samt antallet af artikler den pågældende forfatter har forfattet/medforfattet.

### Opgave 4.4 (5 procent)

Lav en SQL-kommando der henter emailen (`email`) på alle forfattere der IKKE har skrevet nogle artikler.

### Opgave 4.5 (5 procent)

#### *Din Opgave:*

Lav SQL-kommandoen svarende til tabellen fra `list_articles.php` i Figur 1. De hentede rækker skal sorteres efter dato med de nyeste først.

**Bemærk:** Tabellen viser artiklens overskrift (`header`), kategori (`category`), artiklens status (`beingedited`) samt artiklens id `aid` sorteret efter dato (`posteddate`)

## Opgave 5 (30 procent) - Web Service

### Opgave 5.1 (7 procent)

Skriv funktionen `check_login`.

Funktionen tager to strenge som argument. En email og en adgangskode.

Funktionen skal tjekke både email og adgangskode, enten ved hjælp af et regulært udtryk du laver eller ved hjælp af en eller flere af funktionerne fra `utilities.php`

Funktionen skal tjekke om der findes en bruger med pågældende email og adgangskode i tabellen `authors`.

Findes brugeren skal brugerens `aid` returneres.

Findes brugeren ikke skal der returneres `false`.

**Hint:** Der er oprettet forbindelse til databasen og alle nødvendige filer er inkluderet (`include`)

## Opgave 5.2 (7 procent)

### *Din Opgave:*

Lav scriptet `update_article.php`

**Bemærk:** scriptet modtager formvariablerne `header`, `category`, `aid`, `body`, `email` og `pass` fra formen i scriptet; `edit_article.php`.

Scriptet skal tjekke om brugeren findes, evt via `check_login` funktionen.

Findes brugeren valideres alle formvariabler og de(n) relevante rækker i databasen opdateres.

Husk at sætte `beingedited` til 0

Til sidst skal brugeren sendes til `list_articles.php`.

**Hint:** Du skal selv inkludere nødvendige filer.

## Opgave 5.3 (8 procent)

### *Din Opgave:*

Lav scriptet `list_articles.php`

**Bemærk:** Siden modtager ingen formvariable.

Du kan benytte SQL-kommandoen fra 4.5.

Du skal benytte `set_class` fra opgave 2.1 til at sætte rækkernes `class`-attribut

For hver række skal du tjekke om artiklen rettes af en anden (hvis `beingedited` er `TRUE`) og enten lave et link til `edit_article.php` som i opgave 1.2 eller skrive teksten "Artiklen er låst"

**Hint:** Du skal selv inkludere nødvendige filer.

**Hint:** Som i opgave 1.2 behøver du ikke overskrifterne på kolonnerne.

### Opgave 5.4 (8 procent)

#### *Baggrund:*

For at sikre at der ikke er to der retter samme artikel og dermed overskriver hinandens arbejde, benyttes som nævnt feltet `beingedited` i databasen. Denne løsning er dog ikke optimal, da artikler først bliver låst op igen når artiklen bliver gemt.

Begynder brugeren 'a' at rette artiklen og lukker vinduet bliver artiklen aldrig "låst op" igen.

#### *Din Opgave:*

Kom med et eller flere forslag til hvordan ovenstående problematik kunne løses/undgås, mens det grundlæggende, at to ikke kan rette samme artikel samtidigt, bevares.

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