



Høgskolen i Telemark

EKSAMEN

6065 DATABASER OG WEB

09.05.2012

Tid:	9-13
Målform:	Bokmål/nynorsk
Sidetall:	8 med forside og vedlegg
Hjelpemidler:	Ingen
Merknader:	Eksamen blir gjennomført på PC uten tilgang på Internett, men satt opp med nettleser, Notepad++ og WAMP.
Vedlegg:	Syntaks og funksjonsbibliotek

Eksamensresultatene blir offentliggjort på StudentWeb.



Avdeling for allmennvitenskapelige fag



Procentsatsen ut for hver oppgave antyder vekt ved sensur.
Prøv å svare på så mange spørsmål som mulig. Lykke til!

Oppgave 1 (50%)

Tabellene under er tenkt brukt av politiet for å holde rede på hvilke personer og kjøretøy som har vært innblandet i ulykker, der primærnøkler er understreket og fremmednøkler er merket med ei stjerne:

- person(id, fornavn, etternavn, fødselsdato, kjønn)
- kjøretøy(regnr, merke, modell, årsmodell)
- ulykke(unr, dato, vei)
- person_i_ulykke(unr*, id*, rolle, regnr*)

Kolonnene person.id og ulykke.unr er løpenumre som identifiserer henholdsvis personer og ulykker, mens kjøretøy.regnr er standard registreringsnummer. Om hver ulykke blir det lagret dato og på hvilken vei den inntraff. Koblingstabellen person_i_ulykke viser hvilke personer og kjøretøy som har vært involvert i hvilke ulykker. Kolonnen person_i_ulykke.rolle viser hvilken rolle personer hadde, for eksempel sjåfør, passasjer eller fotgjenger. Et SQL-skript tilpasset MySQL som oppretter tabellene og setter inn eksempeldata er lagt inn på hver PC under katalog c:\hit. Forberedelse: Start WAMP, kjør skriptet og gjør deg kjent med tabellene. Du skal løse følgende programmeringsoppgaver:

- Lag et HTML-skjema for å registrere en ny person. Bruker skal skrive inn fornavn, etternavn, fødselsdato og kjønn.
- Lag et PHP-skript som behandler skjemaet fra oppgave 1a. Skriptet skal sjekke at alle felt er utfylt og at kjønn er enten «M» eller «K», og eventuelt gi en fornuftig feilmelding. Hvis inndata er korrekt skal det settes inn en ny rad i tabellen person. Kolonnen id er auto-nummerert. Tips: Uttrykket `empty($_POST['x'])` sjekker om tekstboksen med navn x er utfylt.
- Lag et PHP-skript som viser ulykker i 2010 fordelt på bilmerker. For hvert bilmerke skal det skrives ut antall ganger som et kjøretøy av dette bilmerket har vært involvert i en ulykke. Det er ikke nødvendig å bruke ytre kobling. Utskriften skal presenteres i en HTML-tabell med kolonnenavn på første rad.
- Lag PHP-skript for å knytte personer og kjøretøy til en ulykke. Du kan anta at både ulykken, personer og kjøretøy allerede er registrert i databasen. Brukeren skal skrive inn nummeret på ulykken (unr), samt id, rolle og registreringsnummer på kjøretøy for hver involverte person. Registreringsnummer kan være tomt (f.eks. for en fotgjenger). Informasjonen skal lagres i tabellen person_i_ulykke. For full uttelling skal vilkårlig mange personer kunne registreres.
- Skriv en trigger som sørger for at endring av årsmodellen til et kjøretøy blir loggført i en databasetabell (som du må lage selv). Registreringsnummer, ny og gammel årsmodell skal lagres sammen med informasjon om hvem som gjorde endringen og når det skjedde. Tips: Funksjonen USER leverer navn på innlogget bruker og funksjonen NOW gir nåværende tidspunkt.



Oppgave 2 (25%)

Svar kort på følgende deloppgaver, typisk 3-5 setninger på hver.

- a. Hva er en assosiativ tabell (assosiativ matrise) i PHP?
- b. Hva brukes `$_GET`, `$_POST` og `$_REQUEST` til i PHP?
- c. Hva brukes sesjonsvariabler til i PHP?
- d. Hva skjer ved et kall på funksjonen `mysqli_query` (eventuelt `mysql_query`)?
- e. Hva menes med SQL injection og hvordan kan man sikre seg mot dette?
- f. Hvilke fordeler og ulemper ser du ved lagrede rutiner?
- g. Hva er AJAX?
- h. Hva er forskjellen på et velformet og gyldig XML-dokument?
- i. Hva er en kompleks type i XML Schema?
- j. Hva er ORM?

Oppgave 3 (25%)

Gi utfyllende svar på følgende deloppgaver, 1-2 sider på hver deloppgave.

- a. Forklar hva triggerer er og hva ulike typer av triggerer kan brukes til. Hvilke fordeler og ulemper ser du ved bruk av triggerer?
- b. Hva er en web-tjeneste (web service)? Forklar hvordan WSDL og XML Schema brukes for å definere web-tjenester.



Prosjentsatsen ut for kvar oppgåve antyder vekt ved sensur.
Freist å svare på så mange spørsmål som mogleg. Lykke til!

Oppgåve 1 (50%)

Tabellane under er tenkt nytta av politiet for å holde greie på kva for personer og kjøretøy som har vært innblanda i ulykker, der primærnøklar er understreket og framandnøklar er merka med ei stjerne:

- person(id, fornamn, etternamn, fodselsdato, kjonn)
- kjoretoy(regnr, merke, modell, aarsmodell)
- ulykke(unr, dato, veg)
- person_i_ulykke(unr*, id*, rolle, regnr*)

Kolonnane person.id og ulykke.unr er løpenummer som identifiserer personer og ulykker, mens kjoretoy.regnr er standard registreringsnummer. Om kvar ulykke blir det lagra dato og på kva for veg det skjedde. Koplingstabellen person_i_ulykke syner kva for personer og kjøretøy som har vært involvert i kva for ulykker. Kolonna person_i_ulykke.rolle syner kva for rolle personer hadde, til dømes sjåfør, passasjer eller fotgjengar. Eit SQL-skript tilpassa MySQL som opprettar tabellane og setter inn døme på data er lagt inn på kvar PC under katalog c:\hit. Førebuig: Start WAMP, kjør skriptet og gjer deg kjent med tabellane. Du skal løyse følgjande programmeringsoppgåver:

- a. Lag eit HTML-skjema for å registrere ein ny person. Bruker skal skrive inn fornamn, etternamn, fodselsdato og kjønn.
- b. Lag eit PHP-skript som behandlar skjemaet frå oppgåve 1a. Skriptet skal sjekke at alle felt er utfylt og at kjønn er enten «M» eller «K», og eventuelt gi ein fornuftig feilmelding. Viss inndata er korrekt skal det settast inn ein ny rad i tabellen person. Kolonnen id er auto-nummerert. Tips: Uttrykket empty(\$_POST['x']) sjekkar om tekstboksen med namn x er utfylt.
- c. Lag eit PHP-skript som syner ulykker i 2010 fordelt på bilmerker. For kvart bilmerke skal det skrivast ut kor mange ganger som eit kjøretøy av dette bilmerket har vært involvert i ei ulykke. Det er ikkje naudsynt å nytte ytre kopling. Utskrifta skal presenterast i ein HTML-tabell med kolonnenamn på første rad.
- d. Lag PHP-skript for å knytte personar til ei ulykke. Du kan gå ut i frå at både ulykka, personar og kjøretøy allereie er registrert i databasen. Brukaren skal skrive inn nummeret på ulykka (unr), samt id, rolle og registreringsnummer på kjøretøy for kvar involverte person. Registreringsnummer kan være tomt (til dømes for ein fotgengar). Informasjonen skal lagrast i tabellen person_i_ulykke. For full utteljing skal vilkårlig mange personar kunne registrerast.
- e. Skriv ein trigger som syt for at endring av årsmodellen til eit kjøretøy blir loggført i ein databasetabell (som du må lage sjølv). Registreringsnummer, ny og gamal årsmodell skal lagras saman med informasjon om kven som gjorde endringa og når det skjedde. Tips: Funksjonen USER leverer namn på innlogga brukar og funksjonen NOW gir noverande tidspunkt.



Oppgave 2 (25%)

Svar kort på følgende deloppgåver, typisk 3-5 setningar på kvar.

- Kva er ein assosiativ tabell (assosiativ matrise) i PHP?
- Kva nyttas \$_GET, \$_POST og \$_REQUEST til i PHP?
- Kva nyttas sesjonsvariablar til i PHP?
- Kva skjer ved eit kall på funksjonen mysqli_query (eventuelt mysql_query)?
- Kva meinas med SQL injection og korleis kan ein sikre seg mot dette?
- Kva for fordeler og ulemper ser du ved lagra rutinar?
- Kva er AJAX?
- Kva er forskjellen på eit velforma og gyldig XML-dokument?
- Kva er ein kompleks type i XML Schema?
- Kva er ORM?

Oppgave 3 (25%)

Gi utfyllande svar på følgende deloppgåver, 1-2 sider på kvar deloppgåve.

- Forklar kva trigger er og kva ulike typar av trigger kan nyttas til. Kva for fordeler og ulemper ser du ved bruk av trigger?
- Kva er ein web-teneste (web service)? Forklar korleis WSDL og XML Schema nyttas for å definere web-tenester.



Vedlegg. Syntaks og funksjonsbibliotek

Notatet er en «huskeliste» som tar for seg språkkonstruksjoner og noen vanlige funksjoner brukt i MySQL, PHP og XML. Sidene blir delt ut som vedlegg til eksamen.

MySQL

Lagrede rutiner

```
CREATE PROCEDURE sp_name ([proc-parameter[,...]])  
BEGIN statement_list END
```

```
CREATE FUNCTION sp_name ([func-parameter[,...]]) RETURNS type  
BEGIN statement_list END
```

```
proc-parameter:  
    [ IN | OUT | INOUT ] param_name type
```

```
func-parameter:  
    param_name type
```

Triggere

```
CREATE TRIGGER trigger_name  
    trigger_time trigger_event  
    ON tbl_name  
    FOR EACH ROW  
BEGIN statement_list END
```

```
trigger-time:  
    BEFORE | AFTER
```

```
trigger-event:  
    INSERT | UPDATE | DELETE
```

Markører (cursors)

```
DECLARE cursor_name CURSOR FOR select_statement
```

```
OPEN cursor_name
```

```
FETCH cursor_name INTO var_name [, var_name] ...
```

```
CLOSE cursor_name
```

```
DECLARE action HANDLER  
    FOR condition SET statement;
```

```
action:  
    CONTINUE | EXIT | UNDO
```

```
condition:  
    SQLSTATE value | NOT FOUND
```



Resten av vedlegget inneholder "kodesnutter". Hver linje er uavhengig av de øvrige.

PHP

integer

float

boolean

string

```
$x = 10;
```

```
$t = array(1, 2, 3);
```

```
$x = $t[3];
```

```
$m = array('a' => 1, 'b' => 20, 'x' => 10);
```

```
$y = $m['b'];
```

```
// Kommentar
```

```
/* Kommentar */
```

```
+, -, *, /
```

```
==, ===, !=, <>, !==, <, >, <=, >=
```

```
!, &&, ||
```

```
$s = 'En ' . 'tekst';
```

```
session_start();
```

```
$x = $_SESSION['xyz'];
```

```
session_destroy();
```

```
$x = $_GET['xyz'];
```

```
$x = $_POST['xyz'];
```

```
$x = $_REQUEST['xyz'];
```

```
print("x: $x");
```

```
if (b) { $x=1; } else { ... }
```

```
for ($i=0; $i<10; $i++) { ... }
```

```
foreach ($tab as $elem) { ... }
```

```
foreach ($t as $a => $b) { ... }
```

```
$conn = mysql_connect($host, $user, $passwd);
```

```
$ok = mysql_select_db($db, $conn);
```

```
$result = mysql_query($stmt, $conn);
```

```
$row = mysql_fetch_assoc($result);
```

```
$ok = mysql_close($conn);
```

```
$conn = mysqli_connect($host, $user, $passwd, $db);
```

```
$result = mysqli_query($conn, $stmt);
```

```
$row = mysqli_fetch_assoc($result);
```

```
$ok = mysqli_close($conn);
```



XML, XML Schema og DTD

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<!DOCTYPE xyz SYSTEM "abc.dtd">
```

```
<!ELEMENT a ((b|c)*, (d)+, (e)?)>
```

```
<!ELEMENT b (#PCDATA)>
```

```
<!ATTLIST a b CDATA #REQUIRED>
```

```
<!ATTLIST a c CDATA #IMPLIED>
```

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```
  xmlns="http://www.xyz.com">
```

```
<xsd:complexType name="abc_t">
```

```
<xsd:sequence>
```

```
<xsd:element name="a" type="a_t"
```

```
  minOccurs="5" maxOccurs="unbounded"/>
```

```
<xsd:simpleType name="abc_t">
```

```
<xsd:restriction base="xsd:string">
```

```
<xsd:minInclusive value="10"/>
```

```
<xsd:maxInclusive value="100"/>
```

```
<xsd:maxLength value="10"/>
```

```
<xsd:pattern value="[0-9]{5}"/>
```