



KANDIDATNUMMER:

Eksempel EKSAMEN høsten 2012

EMNENAVN: Databasemodellering og databasesystemer

EMNENUMMER: IMT2571

EKSAMENSdato: 14.12.2012

KLASSE: 10HBSPA, 11HBDRA, 11HBIDATA,
11HBISA, 11HBPUA

TID: 9.00–14.00

EMNEANSVARLIG: Rune Hjelsvold (918 60 974)

ANTALL SIDER UTLEVERT:

TILLATTE HJELPEMIDLER: Ingen

INNFØRING MED PENN.

Ved innlevering skilles hvit og gul besvarelse og legges i hvert sitt omslag.

Oppgavetekst, kladd og blå kopi beholder kandidaten.

Husk kandidatnummer på alle ark.

Les hele oppgavesettet før du begynner og planlegg hvordan du bør bruke tiden for best å besvare hele settet. Beskriv antagelser og fortolkninger du gjør i besvarelsen dersom oppgaveteksten synes uklar.

Oppgave 1 – SQL og relasjonsdatabaser (30%)

Gitt disse tabellene:

STUDENT	Name	StudentNumber	Class	Major
	Smith	17	1	CS
	Brown	8	2	CS

GRADE_REPORT	StudentNumber	SectionIdentifier	Grade
	17	112	B
	17	119	C
	8	85	A
	8	92	A
	8	102	B
	8	135	A

COURSE	CourseName	CourseNumber	CreditHours	Department
	Intro to Computer Science	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Database	CS3380	3	CS

PREREQUISITE	CourseNumber	PrerequisiteNumber
	CS3380	CS3320
	CS3380	MATH2410
	CS3320	CS1310

SECTION	SectionIdentifier	CourseNumber	Semester	Year	Instructor
	85	MATH2410	Fall	98	King
	92	CS1310	Fall	98	Anderson
	102	CS3320	Spring	99	Knuth
	112	MATH2410	Fall	99	Chang
	119	CS1310	Fall	99	Anderson
	135	CS3380	Fall	99	Stone

a) (10%)

Skriv resultatet når følgende spørringer kjøres:

1. `SELECT s.Name, s.StudentNumber
FROM STUDENT s, GRADE_REPORT gr
WHERE gr.Grade = 'B'`
2. `SELECT s.Name, s.StudentNumber FROM STUDENT s
JOIN GRADE_REPORT gr USING (StudentNumber)
WHERE gr.Grade = 'B'`
3. `SELECT s.Name, s.StudentNumber, c.CourseName, COUNT(*)
FROM STUDENT s
LEFT OUTER JOIN GRADE_REPORT gr USING (StudentNumber)
LEFT OUTER JOIN SECTION sec USING (SectionIdentifier)
LEFT OUTER JOIN COURSE c USING (CourseNumber)
GROUP BY s.Name, s.StudentNumber, c.CourseNumber,
c.CourseName`
4. `SELECT COUNT(*)
FROM SECTION
WHERE Year = 98 AND Semester = 'Fall'`
5. `SELECT Name, StudentNumber
FROM Student s
WHERE NOT EXISTS (
SELECT gr.StudentNumber, COUNT(*) cnt
FROM GRADE_REPORT gr
JOIN SECTION sec USING (SectionIdentifier)
GROUP BY gr.StudentNumber, CourseNumber
HAVING cnt > 1)`

b) (10%)

Lag SQL-setninger som:

1. Henter ut kursnummer og kursnavn på alle kurs som har navn som inneholder tekststrengene 'Computer' eller 'Data' – dvs. som for eksempeltabellene ovenfor genererer en tabell som vist på neste side.

<i>CourseNumber</i>	<i>CourseName</i>
CS1310	Intro to Computer Science
CS3320	Data Structures
CS3380	Database

- Henter ut ei fullstendig karakterliste med feltene studentnummer, studentnavn, kursnummer, kursnavn, eksamensår og eksamenskarakter. Lista skal være sortert på studentnummer, eksamensår og kursnummer
- Henter ut en oversikt over hvor mange ganger en emneansvarlig har vært ansvarlig for et emne – dvs. som lager dette resultatet for eksempeltabellene fra forrige side:

<i>CourseNumber</i>	<i>Instructor</i>	<i>Cnt</i>
MATH2410	King	1
CS1310	Anderson	2
CS3320	Knuth	1
MATH2410	Chang	1
CS3380	Stone	1

c) (10%)

Beskriv hva normalisering går ut på og hvilke problemer man kan oppleve dersom en database ikke er normalisert (for eksempel i 3NF/BCNF).

Gitt denne tabellen med tilhørende funksjonelle avhengigheter:

```

Person(id, etterNavn, forNavn, gateAdresse, postnummer,
        poststed, telefoner, ePostAdresser)

id ->      etterNavn, forNavn, gateAdresse, postnummer,
           poststed, telefoner, ePostAdresser
postnummer -> poststed

```

En post i denne tabellen kan for eksempel se slik ut:

```

(301, 'Rune', 'Hjelsvold', 'Ole O Liansv. 24', 2816,
 'Gjøvik', '61135180,61174718,91860974',
 'runehj@hig.no,run-hje@online.no')

```

Er relasjonen i 3NF? Vis hvordan relasjonen kan normaliseres til 3NF dersom den ikke er det.

Oppgave 2 – Databasedesign (30%)

a) (20%)

Anta at du blir ansatt av et universitet for å designe en database for å lagre informasjon knyttet til forelesningsnotater og videoopptak brukt i forelesninger ved universitetet. Forelesninger er knyttet til et bestemt av emnene som tilbys ved universitetet. Emner er identifisert av en emnekode. Foruten emnekode skal databasen holde emnenes tittel, forventet læringsutbytte, en liste av temaer, semester (dvs. høst eller vår), og utdanningssyklus (bachelor-, master- eller doktorgradsnivå). Hvert emne har en ansvarlig faglærer. Faglærer må være en av universitetets fast ansatte forelesere. Forelesere er identifisert med en navnekode (som RuHj, FaAC, JYH, osv.). Databasen skal dessuten lagre forelesernes navn, telefonnumre (kontortelefon og eventuelt mobilnummer), romnummeret på kontoret der de sitter.

Forelesninger opptrer i en ordnet sekvens innen et emne. Databasen skal lagre en kort tittel for hver forelesning og et kort sammendrag. En eller flere forelesere kan opptre i en forelesning. Dessuten kan det opptre gjesteforelesere – dvs. forelesere som ikke har fast ansettelse. Databasen skal lagre navnet til en gjesteforeleser i tillegg til navnet på firmaet han/hun jobber for. Databasen skal lagre URL til eventuelle forelesningsnotater som foreleser eller gjesteforeleser har med seg til en forelesning. Det må være mulig å knytte flere forelesningsnotater til en forelesning, og et forelesningsnotat kan gjerne inngå i flere forelesninger.

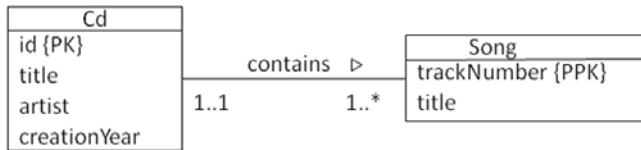
Databasen skal i tillegg lagre informasjon om videoopptak – dvs. tittel på videoopptaket, dato for opptaket og opptakets varighet. Det finnes tre ulike typer videoopptak: ordinære forelesningsopptak, opptak fra gjesteforelesninger og ressursvideoer. Ressursvideoer er videoopptak som en foreleser eller en gjesteforeleser viser i sin forelesning. Databasen skal lagre URL-en til ethvert videoopptak. Opptak fra forelesninger (både ordinære forelesninger og gjesteforelesninger) kan ha åpen (dvs. åpen for alle brukere) eller lukket tilgang (dvs. bare åpen for studenter og ansatte ved universitetet).

Lag en UML-modell for databasen.

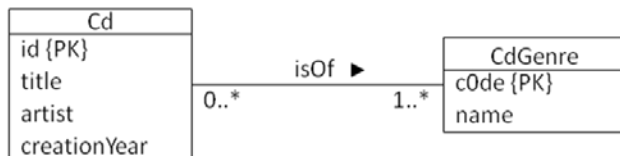
b) (10%)

Gitt de tre UML-konstruksjonene:

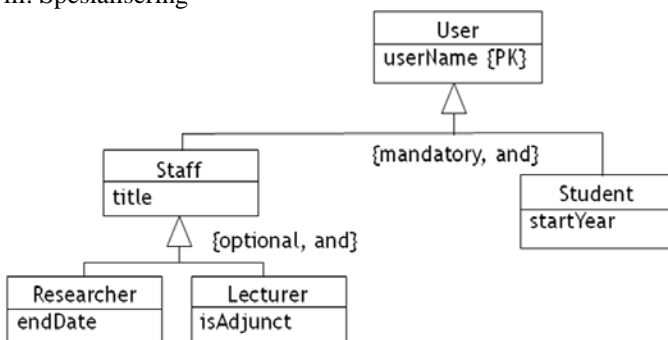
i. Svake enitetstyper



ii. Mange-mange relasjoner



iii. Spesialisering



Lag et logisk relasjonsdesign for hver de tre UML-konstruksjonene vist ovenfor. Begrunn eventuelle valg du tar.

Oppgave 3 – XML (25%)

Eksempel 3.1 viser et XML-skjema for validering av vintester i XML-format.

Eksempel 3.1: XML-skjema for vintester

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Vintest">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Vinliste">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Vinfakta" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Navn" type="xs:string"/>
                    <xs:element name="Aargang" type="xs:gYear"/>
                  </xs:sequence>
                  <xs:attribute name="id" type="xs:string"/>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Beskrivelser" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Vinbeskrivelse" maxOccurs="unbounded">
                <xs:complexType>
```

```

        <xs:sequence>
            <xs:element name="Farge" type="xs:string" />
            <xs:element name="Lukt" type="xs:string"
                maxOccurs="unbounded"/>
            <xs:element name="Smak" type="xs:string"
                maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="vinId" type="xs:string"/>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:key name="vinKey">
    <xs:selector xpath="Vinliste/Vinfakta"/>
    <xs:field xpath="@id"/>
</xs:key>
<xs:keyref name="vinRef" refer="vinKey">
    <xs:selector xpath="Beskrivelse/Vinbeskrivelse"/>
    <xs:field xpath="@vinId"/>
</xs:keyref>
</xs:element>
</xs:schema>

```

a) (15%)

Beskriv strukturen til en vintest som er gyldig i forhold til dette skjemaet. Vis hvordan XML-dokumentet vil se ut for en test av en 2008-årgang av *Lewis Napa Valley Syrah* som har ID 1059907. Vinen skal etter testernes mening ha en dyp purpurfarge med lukt av mørke bær og med smak av kirsebær, bjørnebær og vanilje.

b) (10%)

Anta at vi ønsker å utvide dette XML-skjemaet ved å legge til et nytt attributt kalt `type` med gyldige verdier `rød`, `hvit`, `rose` og `hetvin`. Definer en ny datatype ved bruk av XML-skjema som aksepterer bare disse fire verdiene. Vis hvordan elementet `Vinfakta` må endres for at dette attributtet skal brukes i gyldige vintester.

Oppgave 4 – Andre databasetemaer (15%)

a) (10%)

Beskriv hva en database-cursor er og hvordan den virker. Lag en funksjon i PHP som henter data fra databasen beskrevet i oppgave 1. Funksjonen skal for en gitt student returnere en PHP tabell (array) der elementene er assosiative tabeller med årstall og resultat fra eksamen. Funksjonen skal ha signaturen `getGradeList($sNumber, $dbh)` der `$sNumber` angir studentnummeret og der `$dbh` er et PDO-objekt som er klar til bruk for å hente data fra en database med tabellene. Resultattabellen skal for eksempel se slik ut dersom funksjonen ble kalt med studentnummeret lik 17:

```

array(array("year" => 98, "course" => "MATH2410", "grade" => 'B'),
       array("year" => 98, "course" => "CS1310", "grade" => 'C'))

```

b) (5%)

Hva er SQL Injection? Hva kan vi som utviklere gjøre for å hindre SQL Injection i våre databaseapplikasjoner på web?

Vedlegg A – Utdrag av XML-skjemareferanse

```
< simpleType
id=ID
name=NCName
any attributes
>
  (annotation?,(restriction|list|union))
</simpleType>
```

```
<restriction
id=ID
base=QName
any attributes
>
```

Content for simpleType:

```
(annotation?,(simpleType?,(minExclusive|minInclusive|
maxExclusive|maxInclusive|totalDigits|fractionDigits|
length|minLength|maxLength|enumeration|whiteSpace|pattern)*))
```

Content for simpleContent:

```
(annotation?,(simpleType?,(minExclusive |minInclusive|
maxExclusive|maxInclusive|totalDigits|fractionDigits|
length|minLength|maxLength|enumeration|whiteSpace|pattern)*)?,
((attribute|attributeGroup)*,anyAttribute?))
```

Content for complexContent:

```
(annotation?,(group|all|choice|sequence)?,
((attribute|attributeGroup)*,anyAttribute?))
</restriction>
```

Vedlegg B – Utdrag av PDO referanse

PDO:

[__construct](#) (string \$dsn [, string \$username [, string \$password
[, array \$driver_options]]])
bool [beginTransaction](#) (void)
bool [commit](#) (void)
mixed [errorCode](#) (void)
array [errorInfo](#) (void)
int [exec](#) (string \$statement)
mixed [getAttribute](#) (int \$attribute)
static array [getAvailableDrivers](#) (void)
bool [inTransaction](#) (void)
string [lastInsertId](#) ([string \$name = **NULL**])
PDOStatement [prepare](#) (string \$statement [, array \$driver_options = array()])
PDOStatement [query](#) (string \$statement)
string [quote](#) (string \$string [, int \$parameter_type = PDO::PARAM_STR])
bool [rollBack](#) (void)
bool [setAttribute](#) (int \$attribute , [mixed](#) \$value)

PDOStatement:

bool [bindColumn](#) ([mixed](#) \$column , [mixed](#) &\$param [, int \$type [, int \$maxlen
[, [mixed](#) \$driverdata]]])
bool [bindParam](#) ([mixed](#) \$parameter , [mixed](#) &\$variable
[, int \$data_type = PDO::PARAM_STR [, int \$length
[, [mixed](#) \$driver_options]]])
bool [bindValue](#) ([mixed](#) \$parameter , [mixed](#) \$value
[, int \$data_type = PDO::PARAM_STR])
bool [closeCursor](#) (void)
int [columnCount](#) (void)
bool [debugDumpParams](#) (void)
string [errorCode](#) (void)
array [errorInfo](#) (void)
bool [execute](#) ([array \$input_parameters])
mixed [fetch](#) ([int \$fetch_style [, int \$cursor_orientation = PDO::FETCH_ORI_NEXT
[, int \$cursor_offset = 0]]])
array [fetchAll](#) ([int \$fetch_style [, [mixed](#) \$fetch_argument
[, array \$ctor_args = array()]]])
string [fetchColumn](#) ([int \$column_number = 0])
mixed [fetchObject](#) ([string \$class_name = "stdClass" [, array \$ctor_args]])
mixed [getAttribute](#) (int \$attribute)
array [getColumnMeta](#) (int \$column)
bool [nextRowset](#) (void)
int [rowCount](#) (void)
bool [setAttribute](#) (int \$attribute , [mixed](#) \$value)
bool [setFetchMode](#) (int \$mode)