

# **Datalagring IV1351 HT2017**

## **Kursanvisningar**

<b>1</b>	<b>KURSBESKRIVNING</b>	<b>3</b>
<b>2</b>	<b>LITTERATUR</b>	<b>3</b>
<b>3</b>	<b>LÄRARE</b>	<b>3</b>
<b>4</b>	<b>UNDERVISNINGSFORM</b>	<b>3</b>
<b>5</b>	<b>DATORMILJÖ</b>	<b>4</b>
<b>6</b>	<b>EXAMINATION</b>	<b>4</b>
<b>7</b>	<b>HANDLEDNING</b>	<b>6</b>
<b>8</b>	<b>KURSUTVECKLING OCH KURSUTVÄRDERING</b>	<b>6</b>
<b>9</b>	<b>KURSSCHEMA/FÖRSLAG PÅ ARBETSGÅNG</b>	<b>7</b>
<b>10</b>	<b>ÖVNINGSDATABASES</b>	<b>9</b>
<b>11</b>	<b>LEKTIONSUPPGIFTER</b>	<b>15</b>
<b>12</b>	<b>INTRODUKTIONSUPPGIFTER</b>	<b>22</b>
<b>13</b>	<b>QUIZUPPGIFTER</b>	<b>23</b>
<b>14</b>	<b>LABBUPPGIFTER</b>	<b>23</b>
<b>15</b>	<b>PROJEKTARBETE</b>	<b>25</b>



## **1 Kursbeskrivning**

Kursen ska ge grundläggande kunskap om hur databaser har utvecklats, och om olika datamodeller och hur de används. Kursen ger fördjupade kunskaper i konceptuell modellering och efterföljande relationsdatabasdesign. Kursen tar även upp semistrukturerade databasmodeller och XML. Vidare behandlar kursen frågespråk för relationsdatabaser och XML. Dessutom ger kursen praktisk erfarenhet av konstruktion och hantering av relationsdatabaser (som exemplifieras med hjälp av MS Access och MySQL) och XML. Kursen behandlar också kort databashanteringssystem och mekanismer för bland annat säkerhet. Avslutningsvis orienterar kursen om andra typer av databaser och andra användningsformer av databaser, samt hur man kopplar ihop ett program med databaser.

## **2 Litteratur**

### **Huvudbok**

Database Systems, A Practical Approach to Design, Implementation and Management  
Connolly/Begg, Addison-Wesley/Pearson (Fourth/Fifth/Sixth Edition)  
Läsanvisningar finns i föreläsningsbilderna.

### **Övrigt material**

- Föreläsningsbilder
- Kursanvisningar (det här kompendiet)
- Instruktioner för Etapp 2 av projektarbetet (delas ut senare under kursen)
- Kompendium "Introduction to Microsoft Access"
- Kompendium "MySQL Essentials"
- Kompendium "Querying XML Data with XQuery"
- Annat material (exempeldatabaser, extra övningar, gamla tentor med lösningar, lösningsförslag till lektionsuppgifter)

## **3 Lärare**

nikos dimitrakas (kursansvarig)  
Rum 2423 i Electrum  
Tel: 08-161295  
e-post: nikosd@kth.se

Eventuellt tillkommer några handledare och assistenter.

## **4 Undervisningsform**

Undervisning ges i form av föreläsningar, lektioner, introduktioner, labbar, redovisningar (i samband med projektarbetet), samt genom handledning. I kursen ingår även fyra obligatoriska quizzar som görs elektroniskt. Det är starkt rekommenderat att delta i all undervisning.

Elektronisk handledning och annan kommunikation med lärarna sker via Canvas och via mejl.

## 5 Datormiljö

Under kursen behöver ett antal olika verktyg användas för modellering, databaskonstruktion, och implementering av informationssystem.

Följande verktyg **måste** användas:

- Astah (eller annat verktyg för att rita UML-klassdiagram)
- Dia
- Microsoft Access 2010/2013/2016
- MySQL
- XQuisitor eller BaseX
- Java

Utöver dessa verktyg är det nödvändigt att man har tillgång till Canvas och andra KTH-tjänster som man får tillgång till automatiskt när man är registrerad på kursen.

De flesta verktyg ovan är tillgängliga gratis via webben, eventuellt endast som trials eller endast för icke-kommersiell användning. Flera Microsoft-produkter (inklusive Access) finns tillgängliga för alla studenter via Microsoft Imagine (tidigare kallad Dreamspark) (mer information på KTH:s webbplats). En begränsad version av Astah kan hämtas gratis från ChangeVision:s webbplats. Dia kan hämtas gratis från dia-installer.de. BaseX kan hämtas gratis från basex.org och XQuisitor finns som en färdig jar i Canvas.

## 6 Examination

Kursen har två examinationer. Den ena examinationen är på 3 högskolepoäng och motsvarar tentan. Den andra examinationen är på 4,5 högskolepoäng och består av ett projektarbete, fyra quizzar och en labb. Kursbetyget blir medelvärde av examinationernas betyg avrundat uppåt, dock högst ett steg bättre än tentans betyg.

### TENTA

Salstenta på fyra timmar. Inga hjälpmedel tillåtna (förutom lexikon mellan svenska och främmande språk). Tentan består av tre delar som motsvarar kursens första tre lärandemål. För att få godkänt på tentan måste man uppnå ett visst resultat på varje del och ett visst resultat totalt.

Betygsgränser (vad man måste uppnå på varje del och på hela tentan för att få det betyget på tentan):

	Hela tentan	Per del
A-gräns	94%	87%
B-gräns	85%	75%
C-gräns	77%	64%
D-gräns	68%	52%
E-gräns	60%	40%

Får man minst 75% på en del utan att uppnå E-gränsen, får man rest (Fx) på de delar man inte fick 75% på, dvs möjlighet att endast göra dessa delar vid nästa tenta. Då kan man få högst E. Man behöver inte använda sig av denna möjlighet, men om man använder sig av den gäller E-gränserna för de delar man skriver, dvs 60% på hela tentan och 40% per del.

Det framgår av schemat vilka tentatillfällen som finns planerade just nu. Flera tillfällen kommer att planeras, så kontrollera alltid i schemat.

Gamla tentor (med lösningsförslag) finns tillgängliga i Canvas.

Tentans tre delar täcker följande områden:

**Del 1 (Teori):** Allt om data, databaser, relationsmodellen, normalisering, databashanteringssystem, användning av databaser, embedded SQL, JDBC, etc. Även teori och terminologi om modellering och frågespråk ingår.

**Del 2 (Modellering):** Praktisk modellering, konceptuell och logisk, både för relationsmodellen och för XML.

**Del 3 (Frågespråk):** Praktisk användning av frågespråk, både SQL och XQuery.

### **PROJEKTARBETE**

Ett projektarbete utförs i form av grupparbete, i grupper om tre (3) studenter. Gruppanmälningar görs i Canvas. Använd Canvas för att hitta andra studenter som behöver grupp.

Projektarbetet består av två etapper. Etapp 1 handlar om modellering av en databas utifrån en verksamhetsbeskrivning. Etapp 2 handlar om implementering och användning av en databas inklusive några XML-uppgifter, samt konstruktion av ett enkelt gränssnitt mot databasen i Java. Etapp 1 (modellerna) redovisas iterativt för en lärare vid minst två redovisningstillfällen (ca 15 minuter per tillfälle). Etapp 2 (den färdiga databasen, uppgifterna och Java-programmet) ska lämnas in via Canvas senast kursens sista dag. Dock rekommenderas man göra klart projektarbetet före tentan, gärna redan under december, för att hinna få feedback före tentan.

**OBS: Om gruppen inte är klar med Etapp 1 senast den 1 december 2017, får gruppen inte fortsätta med projektarbetet.** Man får då göra ett nytt projektarbete vid ett senare kurstillfälle.

Redovisningstider för redovisning 1 och 2 tilldelas grupperna utifrån gruppnumret. De exakta tiderna meddelas via Canvas. Det är OK om grupper kontaktar varandra och byter tider. Meddela då även kursansvarig vilka grupper som bytt. Instruktioner för slutinlämningen beskrivs tillsammans med projektarbetets Etapp 2 (som lämnas ut efter tillfredsställande redovisning av Etapp 1). Grupper som inte bedöms kunna gå vidare till Etapp 2 under redovisning 2 kommer att erbjudas nya redovisningstider.

Projektarbetet beskrivs i mer detalj senare i det här kompendiet.

### **QUIZZAR**

Fyra obligatoriska quizzar måste göras individuellt vid bestämda tidpunkter under kursen. Quizzarna görs i Canvas.

- Quiz 1 ska göras efter föreläsning 2. Den täcker det som tas upp under de två första föreläsningarna och den första lektionen.
- Quiz 2 ska göras efter föreläsningarna om SQL och frågespråk. Den täcker det som tas upp på föreläsningarna och lektionerna om SQL och frågespråk.
- Quiz 3 ska göras efter föreläsningen om Embedded SQL i Java. Den täcker det som tas upp på föreläsningen om Embedded SQL i Java och labben. **Quiz 3 ska vara avklarad innan man börjar med labben.**
- Quiz 4 ska göras efter föreläsningen om XML och frågespråk. Den täcker det som tas upp på föreläsningen och lektionerna om XML och frågespråk.

Quizzarna beskrivs i mer detalj senare i det här kompendiet.

**LABB**

En labb ska genomföras och redovisas för handledare framför dator. Man kan jobba ensam eller med hela eller delar av sin grupp eller i annan gruppkonstellation av högst 3. De som har jobbat ihop måste alla närvara vid redovisningen och kunna svara på eventuella frågor. Labben ska redovisas vid något handledningstillfälle under perioden 2017-12-05 – 2018-01-11.

Labben handlar om kopplingen mellan en databas och ett program (i Java). Den är direkt kopplad till föreläsningen om embedded SQL i Java och Quiz 3.

Labben beskrivs i mer detalj senare i det här kompendiet.

## **7 Handledning**

Under kursen ges handledning. Handledningen fungerar som rådgivning runt lösningar (modeller, SQL frågor, etc.) som ni tar fram. Under handledningen kan man också få svar eller förtydliganden på uppgifter eller annat som rör kursinnehållet. Dock ska frågor som rör själva verksamheten som projektarbetet handlar om, ställas till diskussionsforumet "Projektarbetets uppdragsgivare" i Canvas. Handledning ges på plats (i Electrum) och har planerats på de tider som förväntas vara handledningsintensiva. Under dessa tider är handledarna tillgängliga för handledning på plats i Electrum. Man skriver upp sig för handledning via [handledning.dsv.su.se](http://handledning.dsv.su.se). Handledarna bevakar listan elektroniskt och kommer till studenterna så snabbt som möjligt. Man kan sitta var man vill i Electrumbyggnaden: glasgatan, datorsalarna, grupprummen, utanför undervisningssalarna, etc. Handledningstiderna finns i Canvas och kan komma att anpassas utifrån behovet.

Det finns även möjlighet att få handledning elektroniskt. I Canvas finns det möjlighet att ställa allmänna frågor som relaterar till kursinnehållet, administrationen, etc. och frågor som handlar om praktiska problem i projektarbetet. Till exempel kan man ställa frågor om man har problem med ett verktyg eller inte vet hur man gör en viss sak i ett verktyg. Man kan också ställa frågor om satser och konstruktioner i SQL eller XQuery som man inte får att fungera. Här kan man också ställa frågor om den teori som tas upp vid föreläsningar, under lektioner, och i kurslitteraturen. Den elektroniska handledningen fungerar på så sätt att lärare och handledare bevakar Canvas, och svarar så snart de kan. Det är förstås också möjligt för studenterna att svara på varandras frågor, vilket ibland kan betyda att man får svar lite snabbare.

## **8 Kursutveckling och kursutvärdering**

Det är viktigt att man berättar vad man tycker om kursen både under kursens gång och efter kursens slut. Ibland kan det finnas små förändringar som kan göras under kursens gång för att förbättra inläringen eller på annat sätt förbättra kursen. Det enklaste sättet att uppmärksamma oss på synpunkter om kursen är att skriva ett inlägg i Canvas eller skriva direkt till lärarna.

Kursen kan också utvärderas individuellt och anonymt via Canvas direkt efter ordinarie tentan. Utvärderingen är inte till bara i fall man har klagomål. Frågorna är för det mesta flervalsfrågor som kan ge oss en bild av vilka delar av kursen som har mottagits väl och vilka delar som behöver förändras/förbättras. Vi hoppas att varenda deltagare tar tiden att fylla i och skicka utvärderingen.

En analys om kursen sammanställs så småningom med bl a eventuella förslag på förändringar till nästa tillfälle. Kursanalysen publiceras i Canvas. Förändringsarbetet är till stor del baserat på svaren i utvärderingen.

## **9 Förslag på arbetsgång**

Det kan ske förändringar i schemat under kursens gång. För aktuellt schema titta alltid på KTH-webben. Nedan ser man ett förslag på aktiviteter som är lämpliga att utföra vid olika tidpunkter under kursens gång. Kursen är på 7,5 högskolepoäng, vilket innebär 5 veckors heltidsarbete (en arbetsinsats på ungefär 200 timmar). Nedan ser man hur vi har tänkt. Det är dock upp till var och en att finjustera sin planering. Kursen går parallellt med andra kurser så planeringen visar lite grovt vad man bör jobba med i anknytning till de schemalagda aktiviteterna:

### **Vecka 44**

- Föreläsning 1, 2, 3
- Få tag i materialet
- Läs kursanvisningarna
- Ordna alla administrativa aspekter (Canvas, registrering, etc.)
- Läs i boken om Föreläsning 1, 2
- Quiz 1
- Jobba med lektionsuppgifter för Lektion 1
- Lektion 1 (glöm inte anmälan till rätt lektionsgrupp i Canvas)

### **Vecka 45**

- Föreläsning 4
- Läs i boken (kursivt) om Föreläsning 3-4
- Jobba med lektionsuppgifter för Lektion 2-3
- Bilda projektgrupp
- Lektion 2, 3 (glöm inte anmälan till rätt lektionsgrupper i Canvas)
- Introduktion 1
- Experimentera med modelleringsverktyg (Astah, Dia)
- Börja med projektarbetet (konceptuell modell)

### **Vecka 46**

- Jobba med projektarbetet (konceptuell modell)
- Redovisning 1 (konceptuell modell)
- Experimentera med databashanteringssystemen (Access och MySQL) (kompendier)
- Jobba med projektarbetet (revidera enligt redovisningen)
- Redovisning 2 (projektarbetets Etapp 1)
- Föreläsning 5
- Läs i boken om Föreläsning 5-7

### **Vecka 47**

- Föreläsning 6, 7
- Experimentera med databashanteringssystemen (Access och MySQL) (kompendier)
- Jobba med projektarbetet
- Läs i boken om Föreläsning 5-7
- Quiz 2
- Eventuella restredovisningar
- Jobba med lektionsuppgifter för Lektion 4-5

**Vecka 48**

- Jobba med lektionsuppgifter för Lektion 4-5
- Lektion 4, 5 (glöm inte anmälan till rätt lektionsgrupp i Canvas)
- Jobba med projektarbetet
- Eventuella restredovisningar

**Vecka 49**

- Föreläsning 8
- Quiz 3
- Jobba med labbuppgiften
- Redovisa labben
- Jobba med projektarbetet
- Eventuella restredovisningar
- Föreläsning 9
- Läs i boken om Föreläsning 9
- Quiz 4
- Experimentera med XQuery och XQuisitor (kompendium)
- Jobba med lektionsuppgifter för Lektion 6-7
- Lektion 6 (glöm inte anmälan till rätt lektionsgrupp i Canvas)

**Vecka 50**

- Jobba med lektionsuppgifter för Lektion 6-7
- Lektion 7 (glöm inte anmälan till rätt lektionsgrupp i Canvas)
- Föreläsning 10
- Jobba med projektarbetet
- Redovisa labben

**Vecka 51 2017 - Vecka 1 2018**

- Jobba med projektarbetet
- Lämna in projektarbetet (före tentan rekommenderas)
- Kolla gamla tentor
- Repetera inför tentan

**Vecka 2**

- Jobba med projektarbetet
- Lämna in projektarbetet (före tentan rekommenderas)
- Kolla gamla tentor
- Repetera inför tentan
- Lektion 8 (Sammanfattning & repetition inför tentan)
- Tenta
- Fyll i utvärderingen i Canvas

**Senare**

- Omtenta
- Eventuella kompletteringar

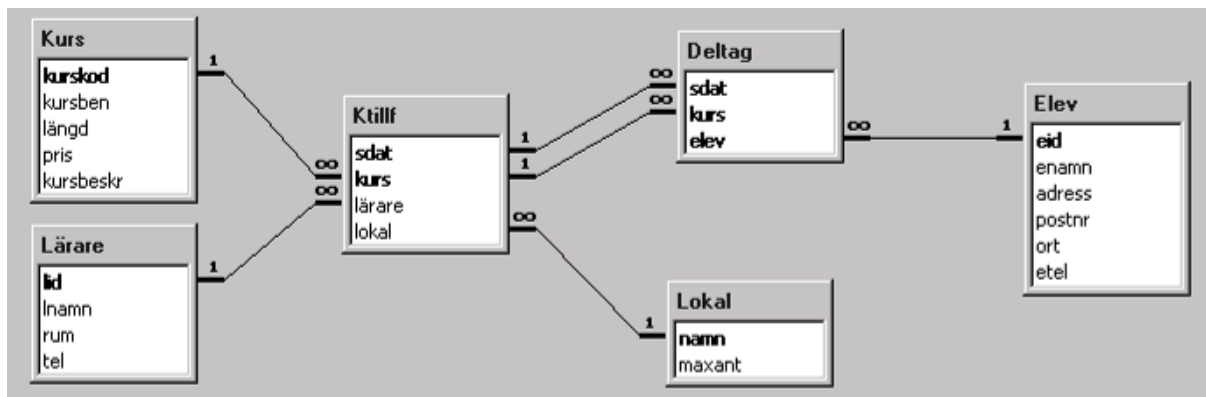


## 10 Övningsdatabaser

Under lektionerna och även för quizzarna kommer följande databaser att användas. Den första databasen är en relationsdatabas om kurser och finns tillgänglig i Canvas som en Access-fil och som en MySQL-dump (ett skript som kan användas för att skapa databasen). Den andra databasen är ett XML-dokument om filmer och finns tillgänglig som en XML-fil i Canvas. I Canvas finns ytterligare övningsdatabaser och övningsmaterial som kan användas på egen hand.

### KURSDATABASEN

#### Grafisk relationsdatabasmodell



Kolumner i **fetstil** utgör tabellens primärnyckel. Relationerna mellan tabellerna betecknar främmande nycklar. T ex är kolumnen lokal i tabellen Ktillf främmande nyckel till kolumnen namn i tabellen Lokal. Kolumnerna sdat och kurs i tabellen Deltag utgör en sammansatt främmande nyckel till Ktillf-tabellens sammansatta primärnyckel.

Det är ganska vanligt att långa tabellnamn måste förkortas i databaser. Tabellen med alla kurstillfällen heter därför Ktillf, medan tabellen med alla deltaganden heter Deltag.

Tabellerna Elev och Lärare har var sin surrogatnyckel (eid respektive lid). Dessa är av datatypen Autonummer. Användaren behöver inte se dem eller jobba med dem.

**Databasinnehåll****Kurs**

<u>kurskod</u>	kursben	längd	pris	kursbeskr
Java1	Java, grundkurs	5	6 700,00 kr	Grundläggande programmering med Java
Java2	Java, fortsättningskurs	4	6 000,00 kr	Avancerad programmering med Java
DBM1	Databasmetodik	2	2 800,00 kr	DB-grunder, normalisering, modellering, SQL
LDBD	Logisk databasdesign	4	6 000,00 kr	Avancerad design, analysmönster, metamodelivering
FDBD	Fysisk databasdesign	5	7 200,00 kr	Lagringsstrukturer, index, denormalisering
Log1	Logik	3	4 500,00 kr	Första ordnings logik, predikatlogik, satslogik

**Ktillf**

<u>sdat</u>	<u>kurs</u>	lärare	lokal
2008-03-06	Java1	2	Tellus
2008-04-02	DBM1	5	Sirius
2008-04-16	Java2	2	Orion
2008-04-16	Log1	3	Jupiter
2008-05-06	LDBD	5	Sirius
2008-05-06	Java1	2	Jupiter
2008-05-09	FDBD	4	Jupiter
2008-09-02	Java1	3	Tellus
2008-09-02	LDBD	4	Orion
2009-01-22	Java2	2	Orion
2009-01-28	DBM1	5	Jupiter

**Lokal**

<u>namn</u>	maxant
Jupiter	12
Orion	24
Sirius	16
Tellus	32

**Lärare**

<u>lid</u>	Inamn	rum	tel
1	Anders Ödman	634	151576
2	Bo Åkerman	604	151526
3	Carl Nordin	603	151553
4	Lena Svensson	605	151556
5	Sofia Wilsson	622	151585

**Deltag**

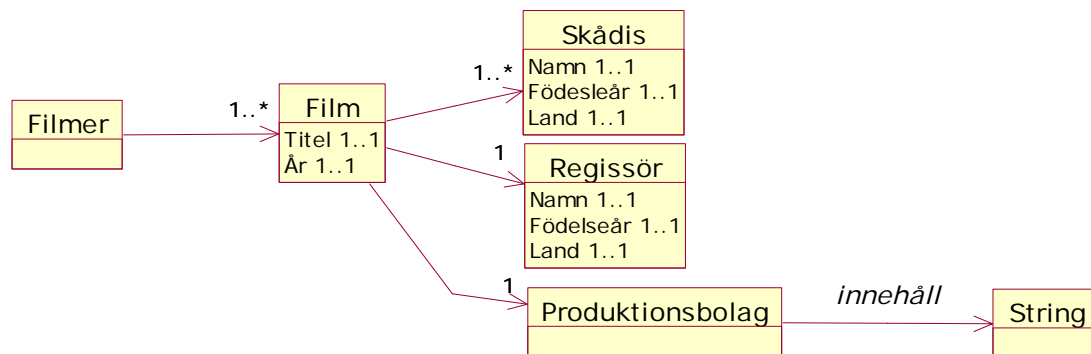
<u>sdat</u>	<u>kurs</u>	<u>elev</u>
2008-03-06	Java1	1
2008-04-02	DBM1	1
2008-04-16	Java2	7
2008-04-16	Log1	4
2008-05-06	LDBD	7
2008-05-06	Java1	2
2008-05-09	FDBD	8
2008-09-02	Java1	1
2008-09-02	LDBD	8
2009-01-22	Java2	2
2009-01-28	DBM1	3
2008-03-06	Java1	9
2008-04-02	DBM1	2
2008-04-16	Java2	5
2008-04-16	Log1	5
2008-05-06	LDBD	1
2008-05-06	Java1	6
2008-05-09	FDBD	6
2008-09-02	Java1	9
2008-09-02	LDBD	6
2009-01-22	Java2	9
2009-01-28	DBM1	5
2008-03-06	Java1	8
2008-04-02	DBM1	10
2008-04-16	Java2	6
2008-03-06	Java1	10
2008-04-02	DBM1	3
2008-03-06	Java1	3
2009-01-22	Java2	10

**Elev**

<u>eid</u>	enamn	adress	postnr	ort	etel
1	Bo Dahl	Ahlgatan 6	16102	Bromma	163578
2	Ann Stål	Lindvägen 3	16429	Kista	373789
3	Ebba Ryd	Ankvägen 4	16107	Bromma	325305
4	Robert Ahl	Ekvägen 1	16425	Kista	123435
5	Lars Holm	Skolgatan 3	16966	Solna	203045
6	Siw Björk	Bokvägen 2	16431	Kista	452678
7	Sigge Ehn	Bokvägen 24	16429	Kista	245578
8	Kurt Grahn	Byvägen 112	19735	Bro	192292
9	Eva Jung	Storgatan 5	16966	Solna	131187
10	Lola Frid	Lillgatan 3	18754	Täby	723384

## FILMDATABASEN

### Grafisk och textuell representation av strukturen



Rotelementet är "Filmer" och det har ett eller flera subelement "Film". Varje element "Film" har attributen "Titel" och "År" och innehåller ett eller flera element "Skådis", exakt ett element "Regissör" och exakt ett element "Produktionsbolag". Varje element "Skådis" och varje element "Regissör" har attributen "Namn", "Födelseår" och "Land". Elementet "Produktionsbolag" har inga attribut eller subelement, men har text som innehåll. Diagrammet ovan motsvarar följande DTD och XML Schema.

Obs! Denna databasmodell har redundans då personer förekommer flera gånger. Detta är avsiktligt för att öva på att hantera redundans korrekt. Under en av lektionerna kommer vi att modellera om denna struktur för att eliminera redundansen.

### XML Schema

```

<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema">
  <element name="Filmer" type="FilmerType"/>
  <complexType name="FilmerType">
    <sequence>
      <element name="Film" type="FilmType" minOccurs="1" maxOccurs="unbounded" />
    </sequence>
  </complexType>
  <complexType name="FilmType">
    <sequence>
      <element name="Skådis" type="PersonType" minOccurs="1" maxOccurs="unbounded" />
      <element name="Regissör" type="PersonType" minOccurs="1" maxOccurs="1" />
      <element name="Produktionsbolag" type="string" minOccurs="1" maxOccurs="1" />
    </sequence>
    <attribute name="Titel" type="string" use="required" />
    <attribute name="År" type="integer" use="required" />
  </complexType>
  <complexType name="PersonType">
    <attribute name="Namn" type="string" use="required" />
    <attribute name="Födelseår" type="integer" use="required" />
    <attribute name="Land" type="string" use="required" />
  </complexType>
</schema>
  
```

**DTD**

```

<ELEMENT Filmer (Film+)>
<ELEMENT Film (Skådis+, Regissör, Produktionsbolag)>
<!ATTLIST Film
    Titel CDATA #REQUIRED
    År CDATA #REQUIRED>
<ELEMENT Skådis EMPTY>
<!ATTLIST Skådis
    Namn CDATA #REQUIRED
    Födelseår CDATA #REQUIRED
    Land CDATA #REQUIRED>
<ELEMENT Regissör EMPTY>
<!ATTLIST Regissör
    Namn CDATA #REQUIRED
    Födelseår CDATA #REQUIRED
    Land CDATA #REQUIRED>
<ELEMENT Produktionsbolag (#PCDATA)>

```

**Databasinnehåll**

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE Filmer>
<Filmer>
  <Film Titel="Driven" År="2001">
    <Skådis Namn="Burt Reynolds" Födelseår="1936" Land="USA"/>
    <Skådis Namn="Silvester Stallone" Födelseår="1946" Land="USA"/>
    <Skådis Namn="Kip Pardue" Födelseår="1976" Land="Canada"/>
    <Regissör Namn="Silvester Stallone" Födelseår="1946" Land="USA"/>
    <Produktionsbolag>Tri-Star</Produktionsbolag>
  </Film>
  <Film Titel="Antz" År="1998">
    <Skådis Namn="Woody Allen" Födelseår="1935" Land="USA"/>
    <Skådis Namn="Silvester Stallone" Födelseår="1946" Land="USA"/>
    <Skådis Namn="Sharon Stone" Födelseår="1958" Land="USA"/>
    <Regissör Namn="Eric Darnell" Födelseår="1961" Land="Ireland"/>
    <Produktionsbolag>Universal</Produktionsbolag>
  </Film>
  <Film Titel="The Muse" År="1999">
    <Skådis Namn="Andie MacDowell" Födelseår="1958" Land="USA"/>
    <Skådis Namn="Jeff Bridges" Födelseår="1949" Land="USA"/>
    <Skådis Namn="Sharon Stone" Födelseår="1958" Land="USA"/>
    <Regissör Namn="Pitof" Födelseår="1970" Land="France"/>
    <Produktionsbolag>October Films</Produktionsbolag>
  </Film>
  <Film Titel="End Game" År="2006">
    <Skådis Namn="James Woods" Födelseår="1955" Land="USA"/>
    <Skådis Namn="Cuba Gooding Jr" Födelseår="1968" Land="USA"/>
    <Regissör Namn="Andy Cheng" Födelseår="1972" Land="China"/>
    <Produktionsbolag>Millennium Films</Produktionsbolag>
  </Film>

```

```
<Film Titel="Picking Up the Pieces" År="2000">
  <Skådis Namn="Woody Allen" Födelseår="1935" Land="USA"/>
  <Skådis Namn="Sharon Stone" Födelseår="1958" Land="USA"/>
  <Skådis Namn="Alfonso Arau" Födelseår="1948" Land="USA"/>
  <Regissör Namn="Eric Darnell" Födelseår="1961" Land="Ireland"/>
  <Produktionsbolag>Tri-Star</Produktionsbolag>
</Film>
<Film Titel="Anything Else" År="2003">
  <Skådis Namn="Woody Allen" Födelseår="1935" Land="USA"/>
  <Skådis Namn="Jason Biggs" Födelseår="1978" Land="USA"/>
  <Skådis Namn="Christina Ricci" Födelseår="1980" Land="USA"/>
  <Regissör Namn="Woody Allen" Födelseår="1935" Land="USA"/>
  <Produktionsbolag>Universal</Produktionsbolag>
</Film>
<Film Titel="Catwoman" År="2004">
  <Skådis Namn="Sharon Stone" Födelseår="1958" Land="USA"/>
  <Skådis Namn="Halle Berry" Födelseår="1966" Land="USA"/>
  <Regissör Namn="Pitof" Födelseår="1970" Land="France"/>
  <Produktionsbolag>Universal</Produktionsbolag>
</Film>
<Film Titel="The Specialist" År="1994">
  <Skådis Namn="Sylvester Stallone" Födelseår="1946" Land="USA"/>
  <Skådis Namn="Sharon Stone" Födelseår="1958" Land="USA"/>
  <Skådis Namn="James Woods" Födelseår="1955" Land="USA"/>
  <Regissör Namn="Eric Darnell" Födelseår="1961" Land="Ireland"/>
  <Produktionsbolag>Tri-Star</Produktionsbolag>
</Film>
<Film Titel="Sliver" År="1993">
  <Skådis Namn="Sharon Stone" Födelseår="1958" Land="USA"/>
  <Skådis Namn="William Baldwin" Födelseår="1963" Land="USA"/>
  <Skådis Namn="Tom Berenger" Födelseår="1949" Land="USA"/>
  <Regissör Namn="Phillip Noyce" Födelseår="1950" Land="Australia"/>
  <Produktionsbolag>Universal</Produktionsbolag>
</Film>
<Film Titel="Last Action Hero" År="1993">
  <Skådis Namn="Sharon Stone" Födelseår="1958" Land="USA"/>
  <Skådis Namn="Arnold Schwarzenegger" Födelseår="1947" Land="Austria"/>
  <Regissör Namn="John McTiernan" Födelseår="1951" Land="USA"/>
  <Produktionsbolag>Universal</Produktionsbolag>
</Film>
<Film Titel="The Saint" År="1997">
  <Skådis Namn="Val Kilmer" Födelseår="1959" Land="USA"/>
  <Skådis Namn="Elisabeth Shue" Födelseår="1963" Land="USA"/>
  <Regissör Namn="Phillip Noyce" Födelseår="1950" Land="Australia"/>
  <Produktionsbolag>Universal</Produktionsbolag>
</Film>
<Film Titel="Hide and Seek" År="2005">
  <Skådis Namn="Robert De Niro" Födelseår="1943" Land="USA"/>
  <Skådis Namn="Dakota Fanning" Födelseår="1994" Land="USA"/>
  <Skådis Namn="Famke Janssen" Födelseår="1964" Land="Netherlands"/>
  <Skådis Namn="Elisabeth Shue" Födelseår="1963" Land="USA"/>
  <Regissör Namn="John Polson" Födelseår="1965" Land="Australia"/>
  <Produktionsbolag>20th Century Fox</Produktionsbolag>
</Film>
```

```
<Film Titel="Demolition Man" År="1993">
  <Skådis Namn="Silvester Stallone" Födelseår="1946" Land="USA"/>
  <Skådis Namn="Wesley Snipes" Födelseår="1962" Land="USA"/>
  <Skådis Namn="Sandra Bullock" Födelseår="1964" Land="USA"/>
  <Skådis Namn="Hiro Kanagawa" Födelseår="1963" Land="Japan"/>
  <Regissör Namn="Marco Brambilla" Födelseår="1960" Land="Italy"/>
  <Produktionsbolag>Warner Bros</Produktionsbolag>
</Film>
<Film Titel="Excess Baggage" År="1997">
  <Skådis Namn="Alicia Silverstone" Födelseår="1976" Land="USA"/>
  <Skådis Namn="Benicio Del Toro" Födelseår="1967" Land="Puerto Rico"/>
  <Skådis Namn="Hiro Kanagawa" Födelseår="1963" Land="Japan"/>
  <Regissör Namn="Marco Brambilla" Födelseår="1960" Land="Italy"/>
  <Produktionsbolag>Warner Bros</Produktionsbolag>
</Film>
<Film Titel="The 6th Day" År="2000">
  <Skådis Namn="Arnold Schwarzenegger" Födelseår="1947" Land="Austria"/>
  <Skådis Namn="Hiro Kanagawa" Födelseår="1963" Land="Japan"/>
  <Skådis Namn="Robert Duvall" Födelseår="1931" Land="USA"/>
  <Regissör Namn="Roger Spottiswoode" Födelseår="1945" Land="Canada"/>
  <Produktionsbolag>Phoenix Pictures</Produktionsbolag>
</Film>
<Film Titel="Cliffhanger" År="1993">
  <Skådis Namn="Silvester Stallone" Födelseår="1946" Land="USA"/>
  <Regissör Namn="Renny Harlin" Födelseår="1960" Land="Italy"/>
  <Produktionsbolag>Warner Bros</Produktionsbolag>
</Film>
<Film Titel="The Kid And I" År="1993">
  <Skådis Namn="Tom Arnold" Födelseår="1959" Land="USA"/>
  <Skådis Namn="Arnold Schwarzenegger" Födelseår="1947" Land="Austria"/>
  <Regissör Namn="Penelope Spheeris" Födelseår="1945" Land="USA"/>
  <Produktionsbolag>Universal</Produktionsbolag>
</Film>
<Film Titel="True Lies" År="1994">
  <Skådis Namn="Tom Arnold" Födelseår="1959" Land="USA"/>
  <Skådis Namn="Jamie Lee Curtis" Födelseår="1958" Land="USA"/>
  <Skådis Namn="Arnold Schwarzenegger" Födelseår="1947" Land="Austria"/>
  <Skådis Namn="Eliza Dushku" Födelseår="1980" Land="USA"/>
  <Regissör Namn="James Cameron" Födelseår="1954" Land="Canada"/>
  <Produktionsbolag>Tri-Star</Produktionsbolag>
</Film>
</Filmer>
```

## 11 Lektionsuppgifter

Kursen innefattar flera lektioner. Här hittar man uppgifterna för dessa lektioner. Man förväntas ha tittat igenom uppgifterna och försökt lösa dem före respektive lektion.

### LEKTION 1 – NORMALISERING

#### Mål

Att förstå och träna på normaliseringen.

Att förstå de vanligaste typerna av redundans och hur man strukturerar en databas så att redundansen blir minimal.

### ÖVNINGAR I NORMALISERING

För uppgifterna nedan normalisera stegvis (1NF, 2NF, 3NF) till högsta möjliga normalform!

#### Uppgift N1

Ett företag som sysslar med utbildning har idag ett manuellt system där verksamheten för varje år fastställs på följande blankett. En blankett för varje kurs.

Kursnr	Kursbenämning		Längd	Pris	
Kursbeskrivning					
Startdatum	Lärare	Rum	Telefon	Lokal	Maxant

För varje lärare anges rumsnummer samt telefonnummer. För varje lokal anges max antal platser i lokalen.

Varje kurs kan hållas ett varierande antal gånger per år. Varje kurstillfälle hålls av en lärare i en lokal.

En lärare håller flera olika kurser varje år.

**Uppgift N2**

Det som är skrivet med **fetstil** förutsätts förtryckt på blanketten.

**AB Snabba Klipp**

**Kundnr:** 02031  
**Namn:** Lisa Nilsson  
**Gatuadress:** Bergvägen 2  
**Postadress:** 348 32 LUND

**Ordernr:** 25123  
**Datum:** 2013-08-24

Artikelnr	Artbenämning	Färg	Antal	A-pris	Belopp
123	Vippla	röd	5	27,00	135,00
124	Vippla	blå	1	29,00	29,00
211	Rysp	vit	1	123,00	123,00
662	Krilk	blå	23	8,00	184,00

**Total:** 471,00

**Moms:** 117,75

**Att betala:** 588,75

**Adress**  
**Garnisonsgatan 234**  
**961 00 BODEN**

**Telefon**  
**0921-98765**

**Bankgiro**  
**6-543-29**

**Uppgift N3****Projektdeltagande**

personnummer 1..1  
projektnummer 1..1  
personnamn 1..1  
telefonnummer 0..1  
projektnamn 1..1  
startdatum 1..1  
slutdatum 1..1  
budget 1..1  
projektuppdrag 1.. \*  
arvode 1..1  
bilregnr 0..1  
bilmärke 0..1  
årsmodell 0..1

**Kommentarer:**

Bilen tillhör personen. Budget avser hela projektet, medan arvode är specifik för varje projektdeltagare.



## **LEKTION 2 & 3 – MODELLERING**

### **Mål**

Att förstå skillnaden mellan en konceptuell modell och en relationsdatabasmodell.

Att träna på konceptuell modellering och relationsdatabasmodellering.

Att träna på översättning av en konceptuell modell till relationsdatabasmodell – att lära sig översättningsreglerna.

### **ÖVNINGAR I MODELLERING**

För uppgifterna M1 till M6 nedan, ta fram en grafisk relationsdatabasmodell genom att först rita en konceptuell modell (i uppgiften M5 är den konceptuella modellen given). Tänk på att relationsdatabasmodellen måste vara i tredje normalform (3NF) och ha primärnycklar (PN), alternativnycklar (AN) och främmande nycklar (FN).

#### **Uppgift M1**

Vid ett sjukhus önskar man ett databssystem för att ha koll på patientbesök. Det som är intressant är att kunna se vilka symptom en patient hade vid besökstillfället. För besöket måste man förstås veta vilken läkare som träffade patienten och när. En läkare kan förstås också drabbas av någon krämpa. En läkare får dock inte undersöka sig själv!

#### **Uppgift M2**

Bokklubben Nytt, Nöje & Kunskap har många medlemmar som man håller information om. Till medlemmarna säljer klubben böcker som de väljer från klubbens webbplats. Varje bok är klassad att tillhöra ett ämne, har ett utgivningsår och ett ISBN-nr. Varje medlem har anmält vilka ämnen (minst ett) de är speciellt intresserade av, så att de ska kunna ges skräddarsydd information. Medlemmarna beställer böcker från klubben till fasta priser. För varje beställning håller man reda på beställaren och datum, och det kan ingå många olika böcker. Bokklubben samarbetar med ett stort antal förlag, men varje bok ges ut av bara ett förlag. Speciellt intressant för bokklubben är att snabbt kunna få fram alla boktitlar i klubbens utbud, som en viss författare har skrivit, antingen själv, eller som medförfattare. OBS: Leveranser och fakturering behöver inte modelleras.

#### **Uppgift M3**

Den här uppgiften är tänkt att illustrera hur man försöker modellera data och inte processer. Vår databas ska stödja den beskrivna processen, men inte modellera processen i sig.

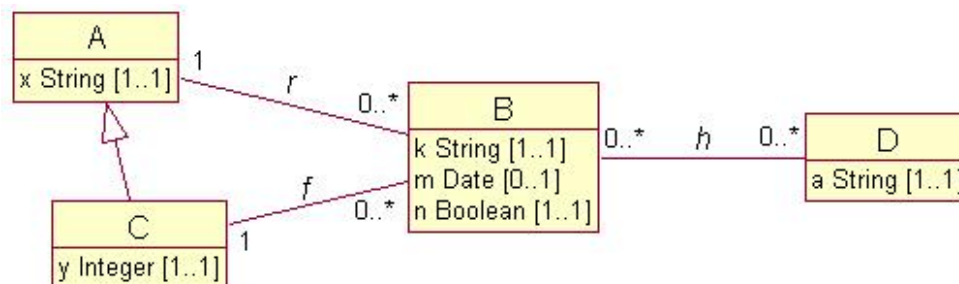
Polisen behöver en databas för att automatisera sin process för framtagning av pass. Det anses särskilt viktigt att kunna spåra när processens delar utförts. Idag fungerar det så här: En person som vill ha pass kommer till polisen och hämtar en ansökningsblankett vid receptionen. Personen fyller i blanketten och lämnar den till passexpeditionen, som fördelar ärendet till en handläggare. Handläggaren tar emot blanketten och kontrollerar att allt är ifyllt korrekt. Sedan måste personen kollas mot brottsregistret och skatter registret. Handläggaren kontaktar därför brottsarkivet och skattemyndigheten för att kolla att de inte har något emot att personen får pass. (Själva brottsregistret och skatter registret behöver inte modelleras, bara att kontrollen gjorts, och utfallet av kontrollen.) Om allt är som det ska enligt brottsarkivet och skattemyndigheten går handläggaren vidare med utfärdandet av själva passet. Innan dess måste dock passexpeditionen jourhavande chef godkänna framtagningen. Så fort passet är klart skickas det med REK-brev till personen.

## Uppgift M4

En kommun vill ha ett gemensamt medlemsregister för alla föreningar i kommunen. Ur systemet ska man kunna få fram personuppgifter för alla medlemmar samt adressuppgifter för alla föreningar. Särskilt önskemål är att dessutom kunna få fram uppgifter om vem som är ordförande i en viss förening.

## Uppgift M5

Översätt den konceptuella modellen nedan till en relationsdatabasmodell. Utöver den grafiska modellen finns också några kommentarer som ska underlätta vissa beslut: ett objekt av klassen B kan identifieras unikt av relationerna r och f tillsammans, dvs inga två B-objekt kan kombineras med samma A-objekt och samma C-objekt.



## Uppgift M6

Nu är det vigsel på gång – igen! Många är också syskonbarnen och kusinbarnen som nog kan tänkas viga sig (och viga om sig!) framöver. Faster Agda, släktens högt älskade kontroll- och teknikfreak, inser nu att en välstrukturerad relationsdatabas är det enda raka för att kunna behålla järngreppet om situationen; hon är fast besluten om att dokumentera alltihop för framtiden.

Men få se nu, vad behövs egentligen för en vigsel? Ja, brud, brudgum och präst är väl självklart! Hmm... men det gäller väl bara heterokyrkovigslar? Bäst är nog att hitta på något mer generellt som kan passa alla sorters vigslar! Platsen för vigseln måste med, liksom den för middagen. Alla gäster ska förstås finnas med, och om dem bör man ha namn och adress, och självklart måste man också lagra vilken relation var och en av dem har till de två i vigselfaret, till exempel 'mormor', 'aerobicspolare', 'skolkamrat', 'brylling', 'WoW-motståndare', osv. Kanske några inte deltar under hela tiden?

Ska middagens olika rätter vara med? Jo, men det kan väl vara roligt att minnas! Men hur ska middagarnas olika delar hanteras? De flesta består säkert av trerättersmenyer med förrätt, varmrätt och dessert, men kanske inte alla? Hursomhelst ska det bli kul att se om det visar sig finnas någon typisk middag!

En sak som är svår att komma ihåg med tiden är presenterna gästerna gav, och förresten kanske inte bara gäster gav? Oftast ger väl varje person en present, men ibland ger någon viss enstaka person, kanske oftast nära släkting, flera presenter, och ibland kanske vissa personer samlar ihop till en gemensam present; det är väl ofta då kompisgäng, men nog inte alltid. Men det borde nog gå att ordna på något smart sätt!

**LEKTION 4 & 5 - SQL****Mål**

Att träna på att hantera SQL för att ta fram information från en relationsdatabas.

Att träna på att använda SQL för att skapa och manipulera en relationsdatabas.

**ÖVNINGAR I SQL**

Uppgifterna nedan ska lösas med SQL. Uppgifterna använder kursdatabasen (som beskrivs i kapitel 10) som kan laddas ner från Canvas för att träna praktiskt med SQL.

**Uppgift S1**

Skapa tabellerna Lokal och Deltag!

**Uppgift S2**

Mata in data i tabellerna Lokal och Ktillf!

**Uppgift S3**

Ändra priset på logikkursen till 5000!

**Uppgift S4**

Höj priset för alla fyraveckorskurser med 10%!

**Uppgift S5**

Ta fram namn och pris på kurser som är på fem veckor!

**Uppgift S6**

Ta bort Lars Holm från logikkursen (dvs ta bort relevanta rader från tabellen Deltag)!

**Uppgift S7**

Ta fram namn på kurser som har gått i lokalen Orion!

**Uppgift S8**

Ta fram namn och telefonnummer på elever som har gått kurs för Sofia Wilsson!

**Uppgift S9**

Ta fram namn och telefonnummer på elever som har haft Carl Nordin eller Sofia Wilsson som lärare!

**Uppgift S10**

Ta fram namn och telefonnummer på elever som har haft Carl Nordin och Sofia Wilsson som lärare!

**Uppgift S11**

Vilken lärare har gett kurser i alla lokaler som rymmer minst 20?

**Uppgift S12**

Vilka lärare har undervisat i Orion men inte i Tellus? Visa lärarens namn!

**Uppgift S13**

Hur många lokaler finns det?

**Uppgift S14**

Hur många kurstillfällen har varje elev gått? Visa namn och antal!

**Uppgift S15**

Hur många kurser har varje elev gått? Visa namn och antal!

**Uppgift S16**

Hur många kurstillfällen har varje lärare hållit? Visa namn och antal! Se till att alla lärare kommer med!

**Uppgift S17**

Vilken lokal har använts flest gånger? Visa namnet!

**Uppgift S18**

Vilken kurs har getts av minst två olika lärare? Visa namn och beskrivning!

**Uppgift S19**

I hur många olika lokaler har grundkursen i Java gått?

**Uppgift S20**

Vilka elever har träffat alla lärare som Ebba Ryd har träffat? Visa namnen!

**Uppgift S21**

Under antagandet att varje elev betalar fullt pris första gången de läser varje kurs och halva pris därefter, vad har varje elev betalat totalt?

**LEKTION 6 & 7 – XML & FRÅGESPRÅK****Mål**

Att förstå hur man designar en XML-struktur utifrån en konceptuell modell.

Att förstå vilka för- och nackdelar en XML-struktur har vad gäller redundans och integritet.

Att träna på att använda XPath-uttryck för att navigera igenom XML-strukturer.

Att träna på att hantera XQuery för att ta fram information från XML-dokument.

**ÖVNINGAR I DESIGN AV XML-STRUKTUR**

Uppgifterna nedan handlar om att designa en lämplig XML-struktur för att representera samma data som en given konceptuell modell. Tänk på starka och svaga entiteter, tänk på integritetsregler som multipliciteterna innebär och tänk på hur man kan undvika onödig redundans! Tänk XML! Strunta i hur modellen skulle översättas till en relationsdatabasmodell. XML har andra möjligheter och begränsningar än vad relationsmodellen har.

**Uppgift XD1**

Skapa en XML-struktur för att representera samma data som den konceptuella modellen av uppgift M1 (från lektionen i modellering)!

**Uppgift XD2**

Skapa en XML-struktur för att representera samma data som den konceptuella modellen av uppgift M2 (från lektionen i modellering)!

**ÖVNINGAR I XQUERY**

Uppgifterna nedan ska lösas med XQuery. Uppgifterna använder filmdatabasen (som beskrivs i kapitel 10) som kan laddas ner från Canvas för att träna praktiskt med XQuery. Uppgifterna kan tolkas lite olika, så prova olika resultatstrukturer.

**Uppgift XQ1**

Ta fram alla regissörer!

**Uppgift XQ2**

Ta fram alla skådespelare från USA som är med i filmen Hide and Seek!

**Uppgift XQ3**

Ta fram antal skådespelare per film!

**Uppgift XQ4**

Ta fram titel på filmer vars regissör är med i filmen som skådis!

**Uppgift XQ5**

Ta fram antal filmer per produktionsbolag!

**Uppgift XQ6**

Ta fram olika skådespelare som Eric Darnell har regisserat!

**Uppgift XQ7**

Ta fram medelåldern av skådespelare i varje film (vid filmens aktuella år)!

**Uppgift XQ8**

Ta fram titel på filmer som Sharon Stone inte är med!

**Uppgift XQ9**

Ta fram filmer med Sharon Stone och/eller Woody Allen!

**Uppgift XQ10**

Ta fram varje skådespelares filmer!

**Uppgift XQ11**

Ta fram alla filmer per produktionsbolag!

**Uppgift XQ12**

Ta fram alla personer och sätt attributet typ till "skådis", "regissör", eller "lite av allt"!

**Uppgift XQ13**

Vad är "Driven", "Pitof", "Universal" och "Canada"? Ta fram element- eller attributnamnet som respektive värde tillhör!

**Uppgift XD3**

Skriv en XQuery-sats för att omvandla XML-dokumentet till ett nytt XML-dokument med följande struktur:

```
<FilmData>
  <Person Namn="" Födelseår="" Land="" />
  <!-- Flera personer -->
  <Film Titel="" År="" Regissör="">
    <Skådis Namn="" />
    <Skådis Namn="" />
    <Produktionsbolag></Produktionsbolag>
  </Film>
  <!-- Flera filmer -->
</FilmData>
```

Alltså en struktur enligt följande DTD:

```
<!ELEMENT FilmData (Person+, Film+)>
<!ELEMENT Person EMPTY>
<!ATTLIST Person
  Namn CDATA #REQUIRED
  Födelseår CDATA #REQUIRED
  Land CDATA #REQUIRED>
<!ELEMENT Film (Skådis+, Produktionsbolag)>
<!ATTLIST Film
  Titel CDATA #REQUIRED
  År CDATA #REQUIRED
  Regissör CDATA #REQUIRED>
<!ELEMENT Skådis EMPTY>
<!ATTLIST Skådis
  Namn CDATA #REQUIRED>
<!ELEMENT Produktionsbolag (#PCDATA)>
```

Spara resultatet som en ny XML-fil! Skriv nu nya XQuery-lösningar för XQ1-XQ13 som fungerar med det nya XML-dokumentet!

## **12 Introduktionsuppgifter**

Under kursen används olika verktyg för modellering med olika notationer och olika databashanteringssystem. Här hittar man övningsuppgifter som kan användas för att bekanta sig med de olika verktygen. Det kan vara bra att göra övningarna innan det blir dags att använda verktygen i projektarbetet.

### **INTRODUKTION 1 – MODELLERINGSVERKTYGEN OCH DATABASHANTERINGSSYSTEMEN**

För MySQL och Microsoft Access kan man även använda sig av de relevanta kompendierna.

#### **Mål**

Att bekanta sig med de verktyg som ska användas under projektarbetet.

Att få en vägledning på hur man bör jobba med projektarbetet.

#### **Uppgift I1**

1. Skapa en enkel konceptuell modell som innehåller lite olika typer av associationer och attribut för beskrivningen nedan.
2. Rita den konceptuella modellen i Astah.
3. Omvandla den konceptuella modellen till en relationsdatabasmodell.
4. Rita relationsdatabasmodellen i Dia.

#### **Uppgift I2**

1. Skapa en databas i MS Access 2013/2016 enligt relationsmodellen från I1.
2. Mata in data och testa olika integritetsregler (PN, FN, datatyper, AN, NULL, etc).
3. Testa att skriva SQL-satser.

#### **Uppgift I3**

1. Skapa en databas i MySQL enligt relationsmodellen från I1.
2. Mata in data och testa olika integritetsregler (PN, FN, datatyper, AN, NULL, etc).
3. Testa att skriva SQL-satser.

#### Verksamhetsbeskrivning:

Det finns personer och kurser. Kurser kan hållas vid olika kurstillfällen. Varje person kan vara student eller lärare, eventuellt både och. Varje kurs har en kursansvarig lärare, och varje kurstillfälle kan eventuellt ha några assistenter, som kan vara antingen studenter eller lärare. Varje student kan delta i flera kurstillfällen, och varje kurstillfälle har flera deltagare. För personerna har vi namn, personnummer och adress. För lärarna har vi deras huvudämne och anställningsnummer. För studenterna har vi deras inskrivningsdatum. För kurserna har vi namn, för kurstillfällena har vi startdatum och slutdatum. Bara ett kurstillfälle kan hållas åt gången per kurs.

## **13 Quizuppgifter**

Kursen innefattar fyra quizzar. Här hittar man instruktioner för dessa. För vissa av quizzarna används kursdatabasen och filmdatabasen som beskrivs i kapitel 10.

### **QUIZ 1 – DATABASGRUNDER, RELATIONSMODELLEN, NORMALISERING**

#### **Mål**

Att repetera teorin från de första föreläsningarna.

#### **Uppgift**

Quiz 1 görs i Canvas. Bör göras före lektion 1.

### **QUIZ 2 – SQL OCH FRÅGESPRÅK**

#### **Mål**

Att repetera teorin från föreläsningarna om frågespråk och SQL.  
Att öva på grundläggande SQL.

#### **Uppgift**

Quiz 2 görs i Canvas. Bör göras före lektion 4.

### **QUIZ 3 – JAVA OCH DATABASER**

#### **Mål**

Att komma igång med Java.  
Att komma igång med MySQL.

#### **Uppgift**

Quiz 3 görs i Canvas. Quizzen ska göras före labben.

### **QUIZ 4 – XML OCH FRÅGESPRÅK**

#### **Mål**

Att repetera teorin från föreläsningen om XML och frågespråk.  
Att öva på grundläggande XPath och XQuery.

#### **Uppgift**

Quiz 4 görs i Canvas. Bör göras före lektion 6.

## **14 Labbuppgifter**

Under kursen ska en labb genomföras och redovisas för handledare. Man kan jobba ensam eller med hela eller delar av sin grupp. De som har jobbat ihop måste alla närvara vid redovisningen och kunna svara på eventuella frågor. Labben kan genomföras när som helst, men ska redovisas under de schemalagda handledningstiderna under perioden 2017-12-05 – 2018-01-15.

### **LABB 1 – EMBEDDED SQL I JAVA**

#### **Mål**

Att bekanta sig med embedded SQL i Java.  
Att prova på att använda en relationsdatabas i ett Java-program.  
Att öva på att använda MySQL.

**Quiz 3 ska göras före labben.****Uppgift**

Med materialet från föreläsningen om embedded SQL i Java som utgångspunkt gör följande:

1. Skapa databasen i MySQL (hämta skriptet från Canvas).  
Skriptet kan köras med kommandot `mysql` (som beskrivs i kompendiet *MySQL Essentials*)  
Alternativt kan man använda Access-versionen av databasen (hämtas från Canvas)
2. Hämta exempelprogrammet, kompilera och kör för att testa att allt fungerar. Tänk på att `mysql-driver-jar`-filen måste finnas tillgänglig när man kör programmet och använder MySQL. På samma sätt måste `UCanAccess`-drivern finnas tillgänglig om man använder Access. (Man kan även välja att köra via ODBC med Java 7 eller tidigare och då måste man skapa ett ODBC-alias).  
Kompilera gör man med kommandot `javac`, t ex `javac MyProg.java`  
Exekvera gör man med kommandot `java`, t ex `java MyProg`  
eller om man vill ta med drivern för MySQL:  
`java -cp .;C:\MySQL\mysql-connector-java-5.1.44-bin.jar MyProg`  
förutsatt att `jar`-filen finns just där. Hittar man inte `jar`-filen, kan man alltid ladda ner den från databashanteringssystemets webbplats eller från Canvas.  
Om man vill köra med `UCanAccess`-drivern får man inkludera `jar`-filerna `ucanaccess-4.0.2.jar`, `commons-lang-2.6.jar`, `commons-logging-1.1.1.jar`, `hsqldb.jar`, `jackcess-2.1.6.jar`, och `ucanload.jar` i `classpath`en. Eller om alla finns i en folder kan man enkelt ange \*:  
`java -cp .;C:\UCanAccess\* MyProg`  
Ändra programmet så att det istället gör följande:
  - a. Visar alla bilmärken.
  - b. Visar alla bilar (regnr, märke och färg) som ägs av någon som bor i en viss stad. Användaren anger en stad.
  - c. Ändrar en bils färg. Användaren ska ange bilens regnr och den nya färgen.
3. Kompilera och kör programmet. Kontrollera att programmet fungerar korrekt.
  - a. Kontrollera att programmet kan utföra alla tre uppgifter (a, b och c)!
  - b. Kontrollera att programmet tar emot rätt input (för b och c)!
  - c. Kontrollera att programmet ger rätt output (för a, b och c)!
  - d. Kontrollera att SQL-satserna gör exakt det som behövs, inte mer, inte mindre!
  - e. Kontrollera att parametrarna hanteras korrekt!
  - f. Kontrollera att variabler och metoder har bra namn!
  - g. En möjlighet är att hitta en annan grupp/student och granska varandras program. Detta är dock frivilligt.
4. Kör programmet och visa/förklara koden för en av handledarna för att få godkänt på labben.

**Frivilliga uppgifter**

1. Modifiera ditt program så att man kan välja databas (MySQL eller Access) när man kör! Låt användaren ange användarnamn och lösenord istället för att hårdkoda i källkoden!
2. Modifiera ditt program så att man har en huvudmeny så att man kan välja att köra a, b eller c, eller avsluta programmet. Då kan man t ex köra b, sedan c, och sedan b igen.
3. Modifiera ditt program så att interaktionen med användaren görs t ex via Java Swing-komponenter (som `JFrame`, `JTextField`, `JButton`, `JLabel`)!



## 15 Projektarbete

### Mål

Att träna på databasmodellering och databasimplementering genom en större uppgift.  
Att förstå rollen av primärnycklar, främmande nycklar, verksamhetsregler, etc.  
Att förstå för- och nackdelar med olika lösningar.  
Att träna på SQL.  
Att förstå skillnader mellan en relationsmodell och en XML-struktur.  
Att träna på XQuery.  
Att förstå skillnaden mellan data i databas och data i användarvyer/användargränssnitt.

### Uppgift

Projektarbetet är uppdelat i två etapper.

#### ETAPP 1

Etapp 1 består av modellering av verksamheten som beskrivs nedan. Etapp 1 redovisas först. Etapp 2 bygger på resultatet av Etapp 1. Etapp 1 redovisas vid minst två tillfällen. *Ta alltid med minst en **utskrift** av modellen/-erna att skriva noteringar på.* Vid den första redovisningen ska följande presenteras och diskuteras med en lärare:

1. en konceptuell modell av verksamheten (ett UML-klassdiagram)
2. en uppsättning verksamhetsregler (2-4 stycken)

Vid den andra redovisningen ska följande presenteras och diskuteras med en lärare:

1. en konceptuell modell av verksamheten (eventuellt reviderad enligt instruktioner av läraren från förra redovisningen)
2. en relationsdatabasmodell utifrån den konceptuella modellen (**skapat i Dia**)
3. en uppsättning verksamhetsregler (eventuellt reviderade och utökade efter förra redovisningen)

Vid redovisningarna får grupperna synpunkter på sina modeller och regler. Om modellerna är bra eller på rätt spår efter den andra redovisningen, får gruppen fortsätta med Etapp 2, annars får gruppen jobba vidare med Etapp 1 och redovisa igen vid ett senare tillfälle. För att modellen ska vara på rätt spår rekommenderas alla grupper att utnyttja handledningen vid minst ett tillfälle före redovisning 1. Av erfarenhet vet vi att grupper som vi inte träffar under handledningen brukar ha mycket sämre modeller vid redovisningarna.

Obs! En bra modell vid ett redovisningstillfälle innebär inte nödvändigtvis att modellen inte behöver förändras ytterligare. Modellen bedöms i detalj efter slutinlämningen. En tillfredsställande modell vid redovisningstillfället innebär att modellens skelett är tillräckligt bra för att gå vidare med. Vår ambition är att gruppen själv under arbetets gång ska kunna upptäcka eventuella förändringar som bör/måste göras i modellen för att den ska bli bättre/funkera.

Ett krav för databasmodellen är att den måste vara i tredje normalform (3NF) och att den inte får innehålla härledbar/beräkningsbar information (redundans).

Utöver verksamhetsbeskrivningen nedan, finns det två extra resurser som kan hjälpa till att belysa eventuella oklarheter i verksamheten och i systemets krav. Verksamhetsbeskrivningen är en allmän beskrivning och allt i denna beskrivning är i princip relevant att fånga i modellen. För att förtydliga kraven på modellen och systemet (vad som måste ingå/hanteras) har vi sammanställt en lista över informationskrav som modellen/databasen måste kunna besvara.

Och om allt detta inte räcker till för att få svar på vad modellen/databasen måste fånga, kan man alltid ställa en fråga till den fiktiva uppdragsgivaren i diskussionsforumet "Projektarbetets uppdragsgivare" i Canvas. Svar i detta forum kommer från uppdragsgivaren (dvs från en person som inte kan så mycket om datorer, databaser och modellering, men som är duktig på sånt som rör verksamheten). **Utnyttja detta forum vid oklarheter om själva verksamhetsbeskrivningen!**

Modellen ska konstrueras **endast** utifrån de beskrivningar som finns nedan och de tillägg som kommer i Canvas. Era kunskaper om liknande verksamheter eller annat i verkligheten ska inte påverka systemet. Det som står nedan är den enda sanningen. Är något otydligt eller oklart kan ni alltid vända er till den fiktiva uppdragsgivaren i Canvas.

**OBS: Om gruppen inte är klar med Etapp 1 senast den 1 december 2017, får gruppen inte fortsätta med projektarbetet.** Man får då göra ett nytt projektarbete vid ett senare kurstillfälle.

## **ETAPP 2**

Etapp 2 innefattar implementering av databasen i MS Access 2010/2013/2016, i MySQL och som XML, samt konstruktion av ett enkelt informationssystem i Java och eventuellt i Access. Även en del frågor ska besvaras med SQL och XQuery. Etapp 2 beskrivs inte i detalj här för att arbetet under Etapp 1 inte ska påverkas. Etapp 2 delas ut till grupperna efter tillfredsställande redovisning av Etapp 1. Vissa av uppgifterna i etapp 2 kommer att vara nödvändiga endast för högre betyg.

Etapp 1 och 2 (dvs modellerna, hela Access-databasen, MySQL-databasen, XML-databasen, SQL-frågorna, XQuery-frågorna, Java-koden, etc.) lämnas in i Canvas. Instruktioner om slutinlämningen (och eventuella restinlämningar) ingår i beskrivningen av Etapp 2.

## **Verksamhetsbeskrivning**

Isa Leglia är VD för företaget spelRvi. Företaget håller på att etablera sig i Sverige och förväntningen är att det ska bli minst 40 fysiska butiker inom ett år. spelRvi kommer att specialisera sig på datorspel med ambitionen att ha Sveriges största sortiment och bra information om alla spel, även de som tillfälligt inte finns i sortimentet. Målsättningen är att man ska kunna nå all information via webben (med bra sökmöjligheter) för att planera sitt köp och sedan vända sig till en butik. Längre fram ska man kunna reservera eller beställa produkter via webben för att sedan hämta i en vald butik, men den biten får vi avvakta med.

## **Spel, plattformar, skapare, produkter och utvecklare**

Det finns datorspel för flera olika plattformar. En plattform är en grundtyp av utrustning som man måste ha för att spela ett visst spel. Det finns tiotals plattformar och de populäraste just nu är Windows, XBOX One, XBOX360, Wii U, Nintendo Switch, Nintendo 3DS, PS3, PS4, PSVita, GameCube. Men det finns förstås många flera och det tillkommer nya plattformar varje år. Varje plattform är tillverkad av ett företag (t ex Microsoft, Sony, Nintendo) och det finns även en liten beskrivning om utrustningen. Det är också viktigt att veta när respektive plattform släpptes på marknaden. Samma spel kan finnas för olika plattformar, men ett visst spel för en viss plattform är en specifik produkt med sin exakta streckkod. Så spelet "Rayman 3: Hoodlum Havoc" som finns för Windows, GameCube, PS3 och XBOX360 leder till fyra olika produkter. Dessa fyra produkter har viss information gemensam. Det är namnet, genrer, innehållsklassificeringen (som beskrivs senare) och en gemensam beskrivning (som gäller oavsett plattform). Varje spel (oavsett plattform) tillhör minst en genre. Det finns ett tiotal genrer

(bl a action, adventure, strategy, racing) och ett spel kan förstås tillhöra flera. Varje version av ett spel (dvs en produkt) är utvecklat av en viss spelutvecklare. Det finns tusentals olika spelutvecklare (t ex Konami, Take-Two Interactive, Flying Iron Games, Ubisoft). Det kan hända att samma spelutvecklare utvecklar flera olika versioner av ett spel, men oftast är det olika spelutvecklare som utvecklar respektive version. Dock finns det alltid en gemensam skapare/designer som står för spelets idé och design oavsett plattform. Denna person eller grupp är spelets skapare (creator). Om skaparen är en grupp (t ex The Brave Ones, TwinGames) måste man ha koll på vilka som ingår i gruppen så att när man söker efter ett spel skapat av en person, ska man även kunna hitta spel skapade av grupper som personen ingår i. Varje produkt släpps vid ett visst datum och släppdatumet brukar vara känt några månader i förväg. För varje produkt måste det finnas möjlighet att ha en beskrivning där man kan ange det som inte framgår av spelets beskrivning och som endast gäller den versionen av spelet (den produkten). Ibland är det onödigt att ha en sådan beskrivning när spelets beskrivning täcker allt för alla versioner.

## **Spelens innehåll**

Utöver information om vilka genrer ett spel tillhör, är det också viktigt att klassificera spelen enligt innehållstyper (Content Descriptors). Dessa innehållstyper beskriver om ett spel innehåller t ex våld, knark, sex, svordomar, etc. Den kompletta listan finns bifogad.

## **Spelserier**

Det är ganska vanligt att spel får uppföljare och då bildas det en spelserie. En spelserie är för en viss plattform och innehåller produkter som hör ihop. T ex Windows-spelen Rayman 1, Rayman 2: The Great Escape, Rayman 3: Hoodlum Havoc, Rayman Raving Rabbids och Rayman Origins tillhör en spelserie. En annan spelserie innehåller alla PS2-spel med Rayman. Och eftersom det finns många Rayman-spelserier, grupperas spelserierna i grupper. Rayman-gruppen skulle alltså bestå av Windows-Rayman-serien, PS2-Rayman-serien, PS3-Rayman-serien, Wii-Rayman-serien, osv. För varje spelserie vill man ha ett namn och en beskrivning.

## **Butiker**

Varje spelRvi-butik kommer att ha sitt eget lager och det är viktigt att dess lagerinformation kan visas på webben, så att man kan se om en produkt finns i en specifik butik. Alla butiker heter spelRvi och då är det adressen eller orten som gör att vi skiljer mellan dem. Om en ort (t ex Stockholm) har flera butiker, måste vi ha koll på adressen, t ex spelRvi på Hamngatan i Stockholm och spelRvi på Sveavägen i Stockholm. För butiker i mindre städer räcker det med stadens namn, t ex spelRvi Övik. Butikerna har dock alltid samma priser (som bestäms centralt). Det är viktigt att informationen som kunderna ser på webben angående pris och tillgänglighet stämmer när de kommer till butiken. I dagsläget finns det inget behov av att spara prishistoriken. För varje produkt är det dock intressant att veta hur många exemplar som någonsin har sålts. Antalet sålda totalt och antalet exemplar per butik behöver hållas uppdaterade i samband med varje köp.

## **Bevakning**

Företaget vill också erbjuda möjlighet att bevaka produkter som inte finns i lager. Man ska kunna ange en epostadress och bevaka en produkt så att man får ett mail när produkten har kommit in till den valda butiken. Så fort produkten har kommit in, skickas mail till alla bevakare och de aktuella bevakningarna tas bort.

## Spelrecensioner

Företaget har undersökt vad som är av intresse för kunderna inför ett köp och resultatet har varit att recensioner är av stor vikt. Därför vill spelRvi ha bra information om vad olika recensenter har gett för betyg till varje produkt. De recensioner som är intressanta är recensioner som spelsajter och speltidningar publicerar. Om recensionen finns online, ska man kunna följa en länk till recensionen. Problemet med recensionerna är att de använder olika skalor, varav de vanligaste är 0-4, 0-5, 0-10, 1-4, 1-5, 1-10, 0-100. spelRvi vill dock presentera ett medelvärde (i procent) och om man vill, ska man kunna se en lista på vilka betyg som har gett detta medelvärde och därifrån ska man kunna gå vidare och läsa respektive recension (om länk finns). Algoritmen för att omvandla ett betyg från en skala till en annan är ganska enkel. Det lägsta möjliga värdet blir 0% och det högsta möjliga värdet blir 100%. T ex en 3:a på en skala 1-4 skulle bli 67%, medan en 3:a på en skala 0-5 skulle bli 60%. Man gör detta för varje recension och sedan tar man medelvärdet.

## Bilder

En stor del av informationen om ett spel är i form av bilder. Det kan handla om bilder på den fysiska produkten (omslaget, skivan, manual), men även bilder från spelet (skärmdumpar). För varje produkt är det viktigt att kunna ha flera sådana bilder och ordningen måste kunna bestämmas så att det finns en vald bild som visas första gången man kommer till en sida om en produkt.

## Informationskrav

Här finns en uppsättning punkter som systemet måste kunna besvara/hantera information om:

- Information om butiker.
- Information om spel, spelserier, grupper av spelserier, produkter.
- Information om plattformar, genrer, spelutvecklare, spelskapare (personer och grupper samt för grupper måste man veta vilka personer som ingår).
- Information om aktuella priser.
- Information om aktuellt lagersaldo och aktuella produktbevakningar i respektive butik.
- Information om spelrecensioner.
- Information om spelinnehållet.

Här finns det också en lista över sådant som systemet inte behöver hantera:

- Personal.
- Försäljning.
- Prishistorik.
- Kunder.

Här finns det också saker som ni kanske känner till från verkligheten som inte ingår i spelRvis verklighet:

- Versioner av samma produkt (t ex Gold edition, Collector's edition, Game of the year edition).
- Spelpaket (t ex två spel i en förpackning, Compilations, Anniversary editions).
- Användarprofiler, inloggningar, kundvagnar, webbshop.
- Information om systemkrav (om processor, grafik, minne, etc)
- Ratinginformation (åldersgräns, ESRB-rating),
- Information om multiplayer, online-gaming, etc.
- Olika språk, eller regionala versioner.

## Content Descriptors

**Alcohol Reference** - Reference to and/or images of alcoholic beverages

**Animated Blood** - Discolored and/or unrealistic depictions of blood

**Blood** - Depictions of blood

**Blood and Gore** - Depictions of blood or the mutilation of body parts

**Cartoon Violence** - Violent actions involving cartoon-like situations and characters. May include violence where a character is unharmed after the action has been inflicted

**Comic Mischief** - Depictions or dialogue involving slapstick or suggestive humor

**Crude Humor** - Depictions or dialogue involving vulgar antics, including "bathroom" humor

**Drug Reference** - Reference to and/or images of illegal drugs

**Fantasy Violence** - Violent actions of a fantasy nature, involving human or non-human characters in situations easily distinguishable from real life

**Intense Violence** - Graphic and realistic-looking depictions of physical conflict. May involve extreme and/or realistic blood, gore, weapons and depictions of human injury and death

**Language** - Mild to moderate use of profanity

**Lyrics** - Mild references to profanity, sexuality, violence, alcohol or drug use in music

**Mature Humor** - Depictions or dialogue involving "adult" humor, including sexual references

**Nudity** - Graphic or prolonged depictions of nudity

**Partial Nudity** - Brief and/or mild depictions of nudity

**Real Gambling** - Player can gamble, including betting or wagering real cash or currency

**Sexual Content** - Non-explicit depictions of sexual behavior, possibly including partial nudity

**Sexual Themes** - References to sex or sexuality

**Sexual Violence** - Depictions of rape or other violent sexual acts

**Simulated Gambling** - Player can gamble without betting or wagering real cash or currency

**Strong Language** - Explicit and/or frequent use of profanity

**Strong Lyrics** - Explicit and/or frequent references to profanity, sex, violence, alcohol or drug use in music

**Strong Sexual Content** - Explicit and/or frequent depictions of sexual behavior, possibly including nudity

**Suggestive Themes** - Mild provocative references or materials

**Tobacco Reference** - Reference to and/or images of tobacco products

**Use of Alcohol** - The consumption of alcoholic beverages

**Use of Drugs** - The consumption or use of illegal drugs

**Use of Tobacco** - The consumption of tobacco products

**Violence** - Scenes involving aggressive conflict. May contain bloodless dismemberment

**Violent References** - References to violent acts