1. *a)* **Regel 1.** *Objektklass som innehåller e-term(er) bildar en egen basrelation*

   Film (Titel, Filmbolag)
   Biograf (Namn, Tel, Platsantal)
   Adress (AdressNr, GatuAdress, GatuNr, Ort)

   Regel 2. *Objektklass som bara består av en i-term och . . .* inte *förekommer på N-sidan i en 1:N-sambandsklass försvinner!*
   ”Plats” och ”DatumTid” försvinner!

   Regel 3. *Sambandsklass av högre ordning än 2 bildar egen basrelation*

   Visning (Titel, DatumTid, Namn)
   Biljett (DatumTid, Namn, PlatsNr, Status)

   Regel 4. *Sambandsklass av typ M:N bildar egen basrelation*

   HarPlats (Namn), PlatsNr)

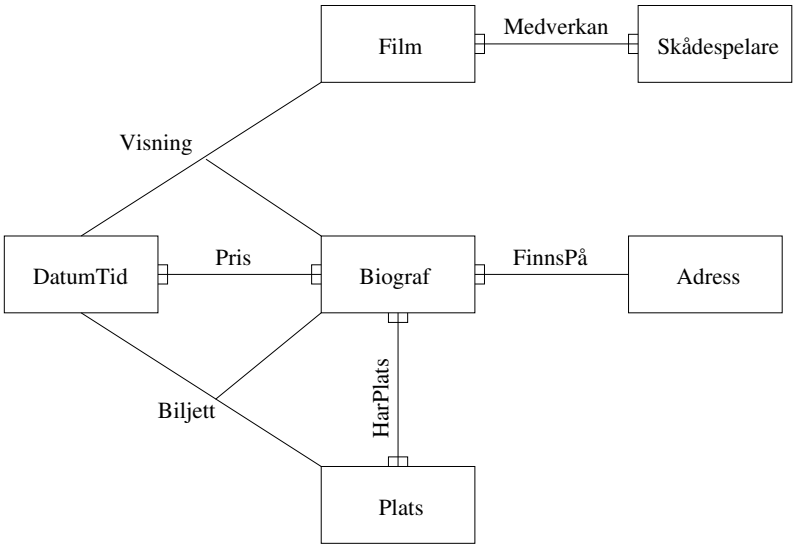
   Regel 5. *Sambandsklass av typ 1:N försvinner men I-termen i objektklassen som finns på 1-sidan blir E-term i objektklassen på N-sidan.*

   Biograf (Namn, Tel, Platsantal, *AdressNr*)

   Slutlig struktur:

   Film (Titel, Filmbolag)
   Biograf (Namn, Tel, Platsantal, AdressNr)
   Adress (AdressNr, GatuAdress, GatuNr, Ort)
   Visning (Titel, DatumTid, Namn)
   Biljett (DatumTid, Namn, PlatsNr, Status)
   HarPlats (Namn), PlatsNr)

   b) En möjlig lösning är att betrakta priset som ett samband mellan en viss biograf och ”datum-tiden” samt att skådespelare representeras i en egen objektklass och har ett samband med en viss film. Då skulle modellen bli:



   Med följande ändringar (tillägg) i egenskapsmatrisen:

Typ	**Namn**	**I-termer**	**E-termer**
Obj	Skådespelare	SNamn	
Samb	Medverkan	Titel, SNamn	Roll
	Pris	DatumTid, Namn	Belopp

2. *a)* ’År’ och ’märke’ bör höra ihop med ’modell’. ’Del’ och ’sort’ hör uppenbarligen ihop och ’kvantitet’ hör ihop med både ’modell’ och ’del’. Baserat på detta ”resonemang” får man tre tabeller:

TVModell	(Modell, År, Märke)
Reservdel	(Del, Sort)
Lager	(Modell, Del, Kvantitet)

   b) Man får

   Insättningsproblem: Man kan t.ex. inte registrera en reservdel utan att ha en modell för den.

   Borttagningsproblem: Om man tar bort en viss TV-modell försvinner alla uppgifter om reservdelar.

   Uppdateringsproblem: Om beskrivningen för en del (’Sort’) ändras måste man gå igenom alla tupler för att ev. uppdatera tupeln.

3. *a)* På vilka våningar finns de avdelningar som har störst antal varutyper i lager?

   b) Vilka varutyper finns till försäljning på de avdelningar som har högre genomsnittslön än sportavdelningen?

Del 2

4.

a) (2p) Content-lenght: behöver vara satt från klienten, så att det är känt hur stor datadelen i POST-requesten är, och Content-Type ska vara application/x-www-form-urlencodedmedia så att servern vet att det är url-kodat data som skickas.

b) (2p) Content-type: behöver vara satt så att mottagaren vet vilken MIME-typ innehållet har, och därmed vet hur det ska behandlas.

5. (2p) Ett envängskrypto genererar alltid samma krypterade resultat C från ett meddelande M. Det krypterade C meddelandet går däremot inte att dekryptera tillbaka till meddelandet M. Det är användbart t.ex. vid lösenordshantering där det räcker att jämföra om ett inmatat lösenords krypterade värde är samma som det som matades in när lösenordet en gång sattes.

6. (2p) Gör en frekvensanalys där man analyserar frekvensen av olika tecken, dels i språket i allmänhet, och dels i den krypterade texten. Tecken som förekommer ofta i den krypterade texten förekommer sannolikt ofta även i språket i allmänhet. Dessutom går det att hitta andra samband. T.ex dubbeltecken är nästan alltid konsonanter.

7. (2p) Ett shell är en kommandotolk. Den kan t.ex. hålla reda på vilka kataloger man ska leta efter exekverbara program, den möjliggör enkla dataflödesstrukturer (loopar, villkor etc) och tillåter att man sätter variabler. Det vanligaste fallet man stöter på det är helt enkelt den kommandoprompt där man skriver in kommandon i unix.

8. (3p) En cookie är en liten fil på klienten där servern kan skriva/läsa några få tecken. För sessionshantering sätter servern en cookie med t.ex. en unik ID på klienten. Vid varje förfrågan klienten gör skickas sedan denna ID med, vilket möjliggör att servern kan lagra ett tillstånd för just denna session (t.ex. i en databas).

9. (3p) OSI-modellens lager: Se föreläsningsanteckningarna.

10. (3p) Kretskopplade nät reserverar en viss bandbredd mellan sändare och mottagare, som vid telefonsamtal. Paketkopplade nät delar istället upp data som ska sändas på paket, som skickas till mottagaren utan någon kontroll över de enskilda paketens färdväg, ungefär som posten. Fördelen med kretskopplade nät är att man har en garanterad bandbredd, nackdelen är dåligt utnyttjande av tillgänglig bandbredd. För paketkopplade nät gäller de motsatta förhållandena.

11. (2p) IP-paket innehåller förutom en IP-adress även ett ”portnummer”. Detta används för att mottagaren ska veta vilken applikation som ska ta emot IP-paketet.

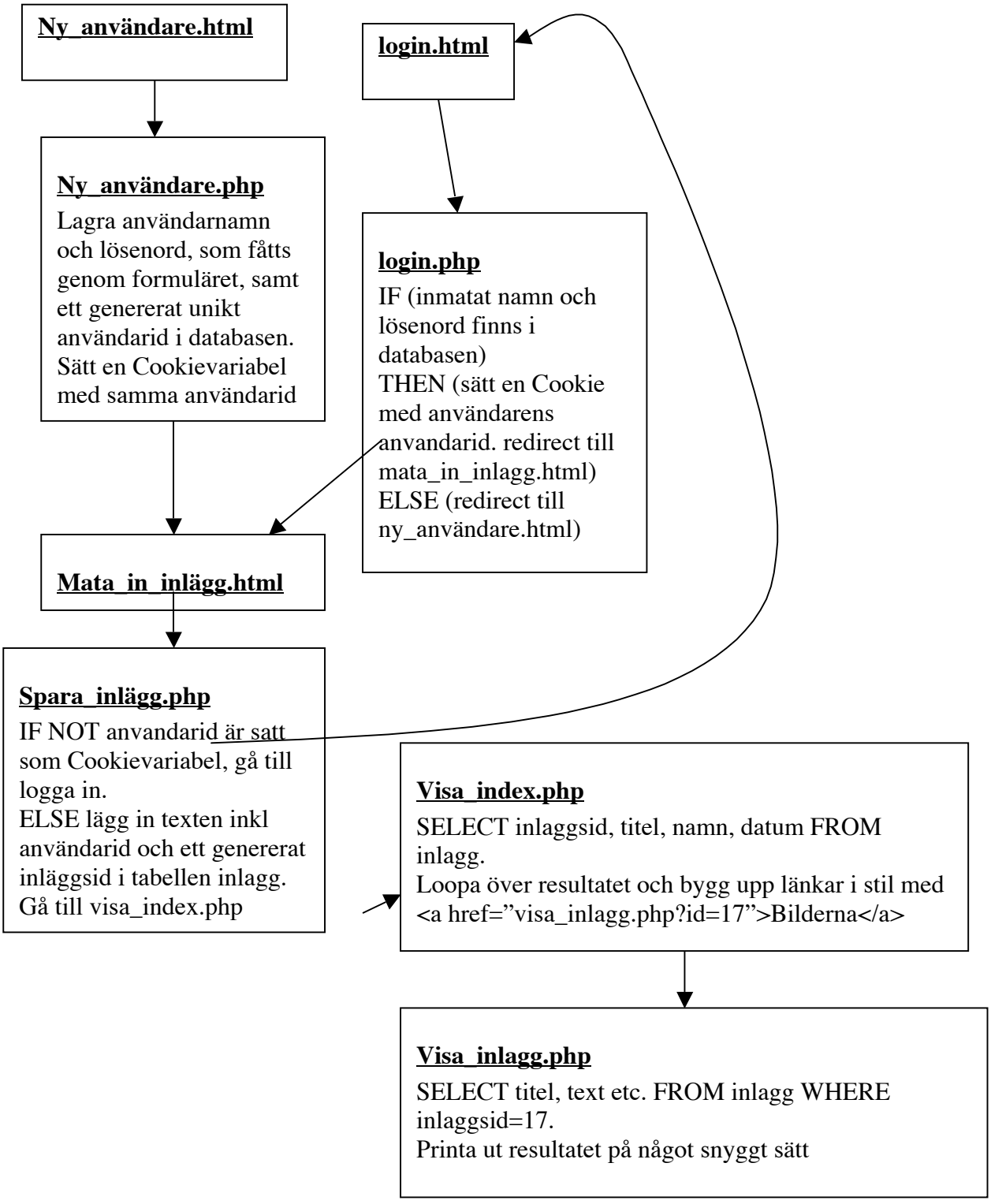
12. (3p) För att ett XML-dokument ska vara ”well formed” krävs att de uppfyller de grundläggande reglerna för hur ett XML-dokument ska vara uppbyggt; Alla tags ska vara avslutade, elementen måste bilda en hirarkisk struktur, alla attribut ska omslutas av citattecken (enkla eller dubbla),tecken såsom < och > ska specialbehandlas.
    För att vara valid krävs att det finns en DTD som dokumentet valideras mot utan fel. DTD:n talar om vilka element som ska vara tillåtna på vilka platser, vilka attribut som ska finnas till respektive element etc.
    ”Well formed” är ett nödvändigt, men inte tillräckligt, villkor för att ett XML-dokument ska vara ”Valid”.

13.

a) (2p)T.ex:
tabellen inlagg: (inlaggid, anvandarid, titel,datum, text)
Tabellen anvandare: (anvandarid, fornamn, efternamn, titel, email)
Anvandarid i respektive tabell utgör en ett-till-mångarelation mellan anvandare och inlagg.

Ovanstående räcker, men om ni ritar T-matriser etc är det givetvis helt korrekt.

b) (6 p) T.ex:


Ny_användare.html
login.html
Ny_användare.php
Lagra användarnamn och lösenord, som fåtts genom formuläret, samt ett genererat unikt användarid i databasen. Sätt en Cookievariabel med samma användarid
login.php
IF (inmatat namn och lösenord finns i databasen)
THEN (sätt en Cookie med användarens anvandarid. redirect till mata_in_inlagg.html)
ELSE (redirect till ny_användare.html)
Mata_in_inlägg.html
Spara_inlägg.php
IF NOT anvandarid är satt som Cookievariabel, gå till logga in.
ELSE lägg in texten inkl användarid och ett genererat inläggsid i tabellen inlagg.
Gå till visa_index.php
Visa_index.php
SELECT inlaggsid, titel, namn, datum FROM inlagg.
Loopa över resultatet och bygg upp länkar i stil med
Bilderna
Visa_inlagg.php
SELECT titel, text etc. FROM inlagg WHERE inlaggsid=17.
Printa ut resultatet på något snyggt sätt

c) (4p) Flera möjliga lösningar, men ett förslag är:

```
<!DOCTYPE forum SYSTEM "forum.dtd">
<forum>
  <inlagg>
    <amne>Bilderna</amne>
    <namn titel="ordforande">
      <fornamn>Olof</fornamn>
      <efternamn>Arnell</efternamn>
    </namn>
    <datum>2002-05-24</datum>
    <text>
      bla bla bla
    </text>
  </inlagg>
  <inlagg>
    <amne>Kommentar</amne>
    <namn>
      <fornamn>tom</fornamn>
    </namn>
    <datum>2002-05-24</datum>
    <text>
      bla bla bla
    </text>
  </inlagg>
</forum>
```

d) (4p) För ovanstående xml-struktur ser DTDn ut:

```
<!ELEMENT forum (inlagg+)>
<!ELEMENT inlagg (amne, namn, datum, text)>
<!ELEMENT amne (#PCDATA)>
<!ELEMENT namn (fornamn, efternamn?)>
<!ELEMENT fornamn (#PCDATA)>
<!ELEMENT efternamn (#PCDATA)>
<!ELEMENT datum (#PCDATA)>
<!ELEMENT text (#PCDATA)>
<!ATTLIST namn titel CDATA #IMPLIED>
```