

Etapp 2 för projektarbete

IV1351 HT2017

spelRvi

Allmänt

Er modell ska implementeras i sin helhet i Access 2010, 2013 eller 2016. I samband med implementationen ska databasen laddas med data. En del av modellen ska implementeras i MySQL och dessa tabeller ska också laddas med data. En del av modellen ska även implementeras som ett XML-dokument med exempeldata. Ett antal frågor ska besvaras i Access med hjälp av SQL. Ett antal frågor ska besvaras med XQuery. Ni ska även skapa en liten del av ett informationssystem i Java och eventuellt i Access för att exemplifiera hur databasen kan användas av slutanvändare.

Implementation i Access 2010/2013/2016

Innan ni börjar med implementationen ska ni ha kört igenom kompendiet "Introduction to Microsoft Access", kapitel 1-5. Kompendiet använder ni sedan också som uppslagsverk under arbetet.

Databasen ska implementeras i sin helhet i Access enligt er modell. Varje tabell ska ha primärnyckel och alla främmande nycklar ska införas liksom andra eventuella regler (alternativa nycklar, NOT NULL, etc) i enlighet med modellen. Obs! Alternativa nycklar implementeras som unika index i Access. Komplexa regler som inte implementeras ska dokumenteras som verksamhetsregler och ni får själva bevaka dem vid datainmatningen.

Tabellerna ska laddas med data med bra variation så att implementationen av modellen kan verifieras och så att resultatet av frågorna nedan kan bedömas. Det är viktigt att testdata följer modellen. Till exempel ska samband som i modellen är ett-till-många avspeglas i testdata. Det innebär att strong entity types behöver minst 3-5 rader, weak entity types minst 8-12 rader, och weak entity types till dem ännu fler. Ingen tabell torde dock behöva ha mer än c:a 40 rader som testdata.

OBS! Samtliga tabeller ska laddas med data. Var kritiska vid inmatningen! Försök att mata in även data som inte borde accepteras av databasen/modellen för att verifiera att databasen beter sig som den ska integritetsmässigt.

Implementation i MySQL

Innan ni börjar med implementationen ska ni ha kört igenom kompendiet "MySQL Essentials". Kompendiet använder ni sedan som uppslagsverk.

En mindre del av modellen ska implementeras i MySQL. Det innebär att ni får välja 6-7 (nära besläktade) tabeller ur er modell och skapa dem i MySQL, samt mata in data i dem. Tabellerna ska implementeras med alla nödvändiga nycklar (primär-, alternativ- och främmande). Endast några få rader behöver matas in i tabellerna, som bevis på att tabellerna och nycklarna fungerar korrekt. Försök även här att mata in felaktiga värden för att kontrollera att databasen beter sig korrekt, det vill säga nekar inmatningen av felaktiga värden.

Frågor (att lösas med SQL i Access)

Frågorna som definieras nedan ska lösas med SQL. De flesta frågorna bör lösas med en SQL-fråga. Endast ett fåtal av frågorna är så komplexa att det möjligen kan vara lämpligt att använda delfrågor (views). Svar som kan innehålla flera rader bör ha minst två rader i svaret. **Noll rader i svaret accepteras aldrig!** Namnge era frågor på ett intuitivt sätt inför bedömningen.

1. Ta fram namn, streckkod och pris på alla produkter som finns i lager i spelRvi Örebro!
2. Ta fram adress på butiker som inte har PS3-spelet "Serious Sam 3: BFE" i lager!
3. Ta fram epostadresser som har bevakningar på minst 3 olika butiker!
4. Ta fram namn och plattform på spelserier som innehåller produkter som ännu inte har släppts!
5. Ta fram antal produkter per spelgrupp!
6. Ta fram streckkod, namn och plattform på alla produkter som har sålt minst 1000 exemplar och som inte ingår i en serie!

Informationssystem (att konstrueras som ett Java-program)

Ett förenklat informationssystem (eller rättare sagt, en liten del av ett informationssystem) ska byggas. Det ska byggas som ett Java-program som använder sig av antingen Access eller MySQL. Systemet ska ha följande funktionalitet:

7. Visar alla produkter.
8. Visar alla butiker som inte har i lager en vald produkt. Användaren ska ange en produkt.
9. Lägger till en ny bevakning i databasen. Användaren ska ange produkt och butik.

Det är tillräckligt om all interaktion med användaren görs via kommandoprompten.

XML och XQuery

Skapa ett XML-dokument med lämplig struktur och data för att lösa följande frågor med XQuery! XML-dokumentets struktur ska designas utifrån den konceptuella modellen och behöver endast täcka de koncept som berör de frågor som ska lösas med XQuery. XML-dokumentet får **inte** genereras automatiskt från t ex Access. Alla XQuery-satser ska returnera ett XML-dokument som svar där rotelementet heter Resultat. Förslag på resultatstruktur på sista sidan.

10. Ta fram alla genrer!
11. Ta fram alla spel per skapare!
12. Ta fram antal produkter per plattform!
13. Ta fram produkter per spelserie!

Alla frågor ska exekveras med XQuery eller BaseX. Se till att ha tillräcklig datavariation i XML-dokumentet för att bekräfta XQuery-satserna!

Betyg

Uppgifterna ovan måste vara godkända för att få betyget E. För högre betyg måste även följande göras:

För betyget D ska allt som krävs för E plus följande göras:

Obs! XML-dokumentet (struktur och data) måste utökas.

SQL:

14. Ta fram namn på skapare som har skapat spel som innehåller "Drug reference" eller "Use of Drugs"!
15. Ta fram antal exemplar som finns på lager av varje produkt i serien Rayman för Windows! Visa spelnamn och antal! Samtliga spel i serien måste ingå i resultatet, även om lagret är tomt.

XQuery:

16. Ta fram information om varje spel! Visa namn, skapare, klassificeringar, genrer och produkterna.
17. Ta fram antal plattformar per spel för Action-spel! Sortera på antalet! Inkludera skaparen!

För betyget C ska allt som krävs för D plus följande göras:

Obs! XML-dokumentet (struktur och data) måste utökas.

SQL:

18. Ta fram epostadresser som har bevakning för samtliga produkter i någon serie! (Obs! inte någon specifik serie)
19. Ta fram streckkod, namn och plattform för produkter som är skapade (direkt eller indirekt) av en viss person! Välj själva en person!

XQuery:

20. Ta fram lagerinformation per butik! Gruppera per butik och per plattform!

XML:

21. Producera ett DTD eller ett XML Schema som motsvarar ert XML-dokument!

För betyget B ska allt som krävs för C plus följande göras:

Java-programmet:

22. All interaktion med användaren ska ske via ett grafiskt Java-gränssnitt (t ex med Java Swing). Funktionalitet beskriven tidigare (7, 8, 9) kan behöva anpassas och utökas. Allt ska hänga ihop så att man får en upplevelse av en applikation.
23. Användaren ska kunna filtrera produkterna genom att välja plattform och/eller genre. Alla plattformar och genrer ska visas så att man kan välja.

SQL:

24. Ta fram namn på spel som har fått (på någon av sina produkter) minst en recension med högsta betyg!

Access:

25. Konstruera ett formulär som kan användas för att bläddra igenom spelgrupper och se information om dessa! Produkterna ska visas grupperade per spelserie.

För betyget A ska allt som krävs för B göras och följande ska göras INDIVIDUELLT:

Obs! XML-dokumentet (struktur och data) måste utökas.

26. Skriv ett XQuery-uttryck som räknar fram recensionernas medelvärde i procent per produkt! Resultatet ska ha ett element per produkt med namn, streckkod, plattform och medelbetygiprocent som attribut. Varje recension ska vara ett subelement med namn, minimum, maximum och betyg som attribut.
27. Skriv en SQL-sats som tar fram alla produkter av en viss genre för en viss plattform rangordnade på medelvärdet av recensionerna! Produkter utan recensioner ska vara med, dock med NULL i kolumnen med medelvärdet. Resultatet ska ha en kolumn med spelnamnet och en kolumn med medelvärdet. Välj själv en genre och en plattform!
28. En fin rapport ska skapas i Access som visar information om ett spel. Visa information om spelet (namn, skapare, genrer, klassificeringar) och information om dess produkter (plattform, släppdatum, utvecklare, pris, antalet sålda, antalet i lager totalt, antalet butiker med produkten i lager).

Inlämning

Inlämning sker via Canvas. **En inlämning per grupp räcker.** Vid felaktig inlämning skriv en kommentar och gör en ny inlämning så snart som möjligt. Det går också bra att kontakta kursansvarig per epost för att få den felaktiga inlämningen raderad. **Gruppnummer och gruppmedlemmarnas namn ska anges i samband med varje inlämning.** Ange även vilket betyg ni har satt som mål i gruppen. Om någon/några i gruppen vill satsa på ett högre betyg än gruppen får denna person/delgrupp göra en extra inlämning. Ange alltid vem som står för inlämningen eller delar av den. Inlämningen ska göras med **en zip-fil** som ska innehålla följande filer:

- En .pdf-fil med er konceptuella modell, er relationsdatabasmodell och eventuella kommentarer och verksamhetsregler.
- En .accdB-fil med hela Access-databasen (med modellen, data, frågorna, rapporterna och formulären).
- En .txt-fil med ett script för hela MySQL-databasen (dvs med kommandon för att skapa tabellerna och för att mata in data) som ni har skapat med kommandot mysqldump.
- En eller flera .java-filer (och eventuellt andra relaterade filer) för ert Java-program.
- En .xml-fil med XML-dokumentet (och eventuellt en .dtd-fil eller .xsd-fil med DTD eller XML Schema).
- En .txt-fil med alla XQuery-uttryck och exekveringsresultatet efter varje XQuery-uttryck.

Om ni har gjort antaganden eller har andra kommentarer som inte framgår i databasen, skriv dessa i meddelandet, eller bifoga en fil med dessa.

Inlämningen ska ske senast terminens sista dag (15 januari 2018).

Rättning

Inlämningarna rättas i den ordning de kommer in. Inlämningen kan antingen bli godkänd eller så får man rest. Om man får rest på E-uppgifterna måste man fixa resten och lämna in igen. Har man fått rest på uppgifter för högre betyg kan man välja att fixa dem eller acceptera det lägre betyget. Vid inlämning av rester gäller samma anvisningar som för den första inlämningen, dock med **deadline en månad efter det att man fick rest.** Restinlämningar ska vara fullständiga. Man kan få rest på rest tills man får godkänt. Vid behov kan gruppen kontaktas för att hålla en demonstration av sitt program. Är programmet välstrukturerat och väldokumenterat, är risken mindre att en demonstration behövs.

Bilaga - Förslag på resultatstrukturer för XQuery

10. Ta fram alla genrer!

```
<Resultat>
  <Genre namn=""/>
  <Genre namn=""/>
</Resultat>
```

11. Ta fram alla spel per skapare!

```
<Resultat>
  <Skapare namn="">
    <Spel namn=""/>
  </Skapare>
  <Skapare namn="">
    <Spel namn=""/>
    <Spel namn=""/>
  </Skapare>
</Resultat>
```

12. Ta fram antal produkter per plattform!

```
<Resultat>
  <Plattform namn="" antalprodukter=""/>
  <Plattform namn="" antalprodukter=""/>
  <Plattform namn="" antalprodukter=""/>
</Resultat>
```

13. Ta fram produkter per spelserie!

```
<Resultat>
  <Spelserie namn="" plattform="">
    <Produkt namn="" streckkod="" pris=""/>
    <Produkt namn="" streckkod="" pris=""/>
  </Spelserie>
  <Spelserie namn="" plattform="">
    <Produkt namn="" streckkod="" pris=""/>
    <Produkt namn="" streckkod="" pris=""/>
    <Produkt namn="" streckkod="" pris=""/>
  </Spelserie>
</Resultat>
```

16. Ta fram information om varje spel! Visa namn, skapare, klassificeringar, genrer och produkterna.

```
<Resultat>
  <Spel namn="" skapare="">
    <Genre namn=""/>
    <Genre namn=""/>
    <Klassificering namn=""/>
    <Klassificering namn=""/>
    <Produkt streckkod="" plattform=""/>
    <Produkt streckkod="" plattform=""/>
  </Spel>
  <Spel namn="" skapare="">
    <Genre namn=""/>
    <Produkt streckkod="" plattform=""/>
  </Spel>
  <Spel namn="" skapare="">
    <Genre namn=""/>
    <Genre namn=""/>
    <Produkt streckkod="" plattform=""/>
    <Produkt streckkod="" plattform=""/>
  </Spel>
</Resultat>
```

17. Ta fram antal plattformar per spel för Action-spel! Sortera på antalet! Inkludera skaparen!

```
<Resultat>
  <Actionspel namn="" skapare="" antalplattformar=""/>
  <Actionspel namn="" skapare="" antalplattformar=""/>
  <Actionspel namn="" skapare="" antalplattformar=""/>
</Resultat>
```

20. Ta fram lagerinformation per butik! Gruppera per butik och per plattform!

```
<Resultat>
  <Butik adress="" stad="">
    <Plattform namn="">
      <Produkt streckkod="" namn="" antal=""/>
      <Produkt streckkod="" namn="" antal=""/>
    </Plattform>
    <Plattform namn="">
      <Produkt streckkod="" namn="" antal=""/>
      <Produkt streckkod="" namn="" antal=""/>
      <Produkt streckkod="" namn="" antal=""/>
    </Plattform>
  </Butik>
  <Butik adress="" stad="">
    <Plattform namn="">
      <Produkt streckkod="" namn="" antal=""/>
      <Produkt streckkod="" namn="" antal=""/>
      <Produkt streckkod="" namn="" antal=""/>
      <Produkt streckkod="" namn="" antal=""/>
    </Plattform>
  </Butik>
</Resultat>
```

26. Skriv ett XQuery-uttryck som räknar fram recensionernas medelvärde i procent per produkt! Resultatet ska ha ett element per produkt med namn, streckkod, plattform och medelbetygiprocent som attribut. Varje recension ska vara ett subelement med namn, minimum, maximum och betyg som attribut.

```
<Resultat>
  <Produkt namn="" streckkod="" plattform=""
    medelbetygiprocent="">
    <Recension namn="" min="" max="" betyg=""/>
    <Recension namn="" min="" max="" betyg=""/>
  </Produkt>
  <Produkt namn="" streckkod="" plattform=""
    medelbetygiprocent="">
    <Recension namn="" min="" max="" betyg=""/>
    <Recension namn="" min="" max="" betyg=""/>
    <Recension namn="" min="" max="" betyg=""/>
    <Recension namn="" min="" max="" betyg=""/>
  </Produkt>
</Resultat>
```