

Institutionen för

TENTAMEN

Kurs: Databaskonstruktion

Examinationsmoment: Salstentamen

Kurskod: IT429G

Högskolepoäng: 2.5hp

Datum: 2025-10-27

Tentamenstid: 14:15 – 19:30

Ansvarig lärare: Johan Bjurén

Berörda lärare

Hjälpmedel/bilagor

Övrigt

Anvisningar

- ☐ Ta nytt blad för varje lärare
- ☒ Ta nytt blad för varje ny fråga
- ☒ Skriv endast på en sida av papperet.
- ☒ Skriv namn och personnummer på samtliga inlämnade blad.
- ☒ Numrera lösbladen löpande.
- ☒ Använd inte röd penna.
- ☒ Markera med kryss på omslaget vilka uppgifter som är lösta.

Poänggränser

Skrivningsresultat bör offentliggöras inom 18 arbetsdagar

Lycka till!

Antal sidor totalt

Tentamen

Databaskonstruktion

Tentamen består av 3 uppgifter som täcker kursens delmoment: 1. Databasimplementation, 2. Applikation PHP och 3. Applikation MVC. Alla delmoment utgår från en angiven verksamhetsbeskrivning (Case) där dina svar ska besvaras mot.

För att erhålla ett godkänt betyg på tentamen måste samtliga 3 delmoment och alla dess uppgifter vara minst godkända. För betyg väl godkänd på tentamen måste minst 2 uppgifter vara besvarade och uppfylla VG uppgiften på samtliga 3 delar.

Svaret på en uppgift bedöms godkänt om svaret på frågan är korrekt bortsett från små felaktigheter. För VG-uppgiften krävs att svaret är i stort korrekt utan felaktigheter och utförligt beskrivet.

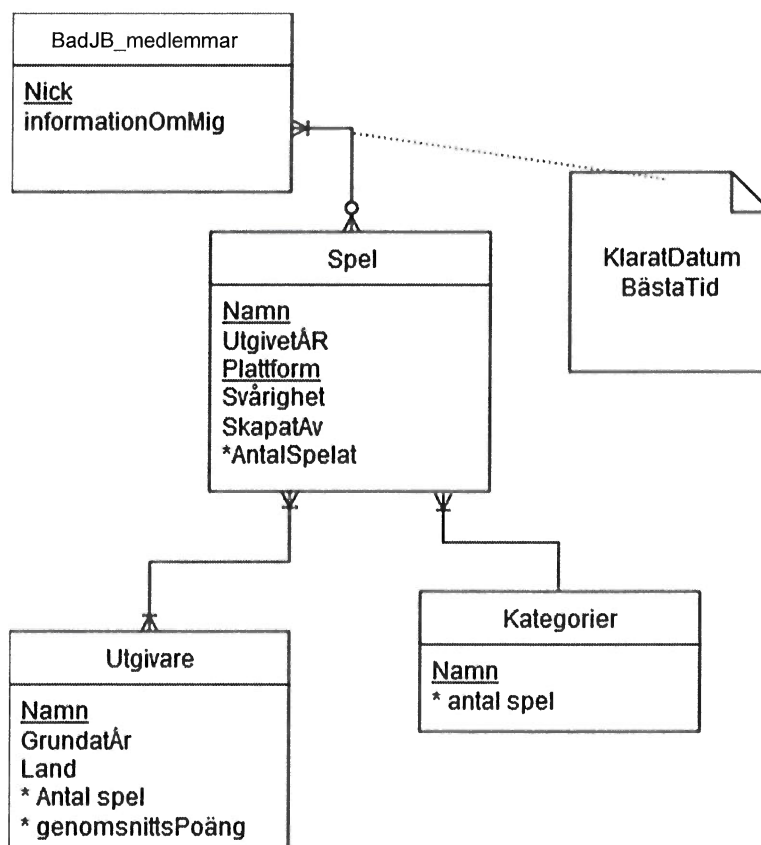
Ingen av frågorna kräver ett svar med programkod. Men skulle det underlätta för ditt svar så är det tillåtet.

Case: Tv-spelssamlarna

Bad J. Bjurrens är en hyperproffsionell organisation som tar extremt seriöst på tv-spel. Utöver att de strävar efter att ha en komplett databas över alla tv-spel som någonsin gjorts så vill de även ha koll på när spelen gavs ut, vem som är utgivare och vilka som gjort spelet. De har även med en 5 gradig skala på hur svårt ett spel är och ett betyg hur roligt ett spel är på en skala 0 till 23. Ett spel identifieras av sitt namn, men eftersom det kan finnas på flera olika plattformar (PC, MAC, XBOX, Playstation...) och man vill även ha koll på hur många som har spelat det specifika spelat av sina medlemmar.

Utöver detta så har de koll på vilka inom organisationen som spelat spelet, vilken tid de klarat spelet på (för att ha koll på så kallade speedrunners inom organisationen), vem som först officiellt klarade spelet.

Bad J. Bjurrens medlemmar har möjligheten att vara anonyma och man sparar endast smeknamn (nick) och ett större textfält där medlemmarna kan beskriva sig själva om dom vill.



Uppgift 1: Databasimplementation (Frågor A-C)

- A.** Välj ett ställe utifrån CASE:et där du kan göra en merge som kan leda till optimering. Rita upp en modell likt den i CASE:et för hur tabellen kan se ut efter en merge, diskutera varför och hur din merge leder till optimering

VG: När du gjort din merge enligt ovan diskutera hur nycklar påverkas samt ta även upp vilka typer av frågor som skulle gynnas av en sådan merge, ge minst ett exempel utifrån CASE:et.

- B.** Beskriv utförligt vad som kännetecknar vertikal denormalisering (vertical split) samt horisontell denormalisering (horizontal split). På vilket sätt skulle vertikal och en horisontell splitt kunna öka prestandan när information efterfrågas - ge exempel utifrån CASE:et.

VG: Utgå från tabellen du tar upp i fråga **A**), Tabellen är en merge av 2 tabeller, visa hur tabellen som du föreslår skulle se ut efter att du har implementerat en vertikal eller horisontell split, visa med ny modell eller create sats, förklara även hur nycklar skulle påverkas.

- C.** Stored procedures och triggers är två olika tekniker som använts flitigt under kursen. Ange vad som kännetecknar stored procedures och triggers samt vad som skiljer sig mellan stored procedures och triggers. Beskriv även varsitt exempel utifrån CASE:et där dessa tekniker kan användas. Motivera ditt svar utförligt.

VG: En av dessa tekniker kan användas för att försvåra för så kallade sql-injections, vilken och på vilket sätt kan den försvåra?

Uppgift 2: Applikation PHP (Frågor A-C)

I denna uppgift ska ni svara utifrån det vi gjort i PHP uppgiften och de bibliotek vi använt i kursen.

- A.** I kursen har vi lärt oss att ansluta till en databas med hjälp av Prepared statements(PDO). Med PDO kan man skicka in värden i en SQL fråga med hjälp av en parameter. Vad får vi för fördelar i jämförelse med att inte använda oss av PDO med att skicka in data på detta viset? Ge ett exempel på en fråga från CASE:et där vi med fördel använder oss av PDO.

VG: Vi har tagit upp att man kan få ange parametrar i uppkopplingen av PDO, Diskutera hur det kan påverka vår kod om vi ändrar mode i PDO från FETCH_BOTH som är default till FETCH_NUM eller FETCH_ASSOC

- B.** Vi har i kursen tittat på att skicka och ta emot data på två olika sätt med hjälp av PHP. Beskriv vilka dessa sätt är och vilka för och nackdelar de har. Ge exempel utifrån CASE:et när det och om det finns någon fördel att skicka på ett visst sätt, ge exempel på.

VG: Istället för att skicka data mellan 2 olika sidor så har vi material i kursen för hur man kan använda sig av sessions för att transportera data från en sida till en annan för en användare. Beskriv skillnaden på att transportera data från en sida till en annan med hjälp av sessions i jämförelse med att formulär.

- C.** Vi har haft exempel på hur man kan arbeta med formulär på samma sida som vi får vår output från formuläret på (singlepage application) och när vi har olika sidor för formulär och sidan som tar emot och ger output från formuläret (form-response). Beskriv skillnaden på dessa, vad behöver du göra olika när du implementerar dessa sidor med tanke på input och output av data?

VG: Diskutera hur strukturen på dokumentet, dvs var du placerar din kod i dokumentet och hur det påverkar hur din output i en single page application.

Uppgift 3: Applikation MVC (A-C)

I denna uppgift ska ni svara utifrån C# uppgiften och de bibliotek vi använt i kursen. Ni ska, som i övriga delar av kursen, svara med den MVC struktur vi tagit upp och använt i kursens andra delmoment.

- A.** Förklara hur flödet i MVC applikationen från att en användare matar in data i ett formulär till att det läggs in i databasen, för att sedan presenteras på webbapplikationen tillsammans med resten av datan i tabellen som datan matades in i. Se till att du tar upp alla 3 delar som MVC består av. Skriv även en kort reflektion hur MVC strukturen skiljer sig från den procedurella programmeringen som vi utförde i PHP.

VG: Utgå från ditt exempel ovan, illustrera och förklara med text ett exempel på dataflödet utifrån CASE:et.

- B.** Beskriv hur kopplingen mellan kontroller och modell ser ut för att lägga in en ny rad(tupel). Ge ett exempel utifrån en tabell från CASE:et med data från ett antal text-inputs. Hur får vi över datan från vår kontroller till vår modell?

VG: I vår modell har jag tagit upp att vi kan ha metoder som har returtypen void och andra DataTable, Förklara skillnaden på dessa samt Ge ett exempel utifrån CASEt när någon av dessa skulle kunna användas.

- C.** Förklara utifrån koden under vad de **fetmarkerade** raderna gör. Ge ett exempel på vad jag skulle kunna få ut för resultat om jag matar in ett "J" exempelvis om det fanns 4 kunder i databasen med namnen "Johan", "Kalle", "Börje" och "Jessica"

VG: Förklara hur jag skulle kunna förändra koden nedan så att den kan hantera sökningar i från två olika tabeller i databasen från samma textinput.

```
public DataTable SearchCustomers(string name)
{
    MySqlConnection dbcon = new MySqlConnection(_connectionString);
    dbcon.Open();
    MySqlDataAdapter adapter =
        new MySqlDataAdapter("SELECT * FROM CUSTOMER WHERE name LIKE @NAME;", dbcon);
    adapter.SelectCommand.Parameters.AddWithValue("@NAME", "%" + name + "%");
    DataSet ds = new DataSet();
    adapter.Fill(ds, "result");
    DataTable CustomerTable = ds.Tables["result"];
    dbcon.Close();
    return CustomerTable;
}
```