

Laborationer - databaser

EDAA20 Programmering och databaser

I kursens databasdel ingår två obligatoriska laborationer.

- Laboration 1 ger träning på att använda SQL för att hämta information från databaser. Den är uppdelad på två tillfällen.
- Laboration 2 handlar om att modellera data i UML.

Uppgifterna i laborationerna ska lösas enskilt eller i grupper om två personer. Se riktlinjer för laborationsuppgifter på kursens hemsida.

Laborationerna är obligatoriska. Kontakta kursansvarig om du skulle vara sjuk vid något tillfälle.

EDAA20 Programmering och databaser, godkända laborationsuppgifter

Skriv ditt namn och din namnteckning nedan:

Namn:

Namnteckning:

Godkänt moment	Datum	Laborationshandledarens namnteckning
Lab 1 (SQL)		
Lab 2 (UML)		

För att bli godkänd på en laboration måste du ha löst uppgifterna i laborationshäftet:

- SQL-frågorna ska ha provkörts i databashanteraren och producerat förväntat resultat. De olika SQL-frågorna ska sparas med unika namn.
- Övriga frågor/uppgifter ska vara korrekt besvarade/lösta och nedskrivna på papper eller i labhäftet. Du ska ha diskuterat dina svar med någon av labbledarna.

Se till att labhandledaren noterar dina laborationsuppgifter som godkända på detta godkännandeblad. Dessa underskrifter är ditt kvitto på att du är godkänd på laborationerna i fall det skulle bli något fel i institutionens noteringar.

Laboration 1 – SQL

Under denna laboration ska du träna på att skriva SQL-frågor samt på att använda en databashanterare.

Mål

Exempel på vad du ska kunna efter laborationen:

- Förklara begreppen relation (tabell), attribut (kolumn), tupel (rad).
- Söka i databasen genom att skriva frågor i SQL:

```
SELECT ...  
FROM ...  
WHERE ...;
```

- Sortera (ORDER BY ...).
 - Döpa om kolumner (AS ...), slå samman kolumner (... + ...).
 - Matcha strängar (LIKE ...).
 - Undvika att visa dubletter (DISTINCT).
 - Använda aggregeringsfunktioner (SUM(...), AVG(...), MIN(...), MAX(...), COUNT(...)).
 - Söka i flera tabeller (JOIN).
 - Använda gruppering (GROUP BY).
 - Använda subfrågor.
- Använda en databashanterare för att skapa databaser och ändra i tabeller (sätta in tupler, ta bort tupler, uppdatera värden).

Förberedelser

- Läs bilderna från första databasföreläsningen.
- Läs avsnittet Bakgrund.
- Skriv SQL-frågor för uppgift 3 a-d på papper eller i labbhäftet.

Laborationen går över två tillfällen. För att vara i fas kan du ha som riktlinje att hinna med uppgift 1–7 under första laborationstillfället. Uppgift 8 passar bra som förberedelse inför andra labbtillfället.

Bakgrund

Vi ska arbeta med en databas som innehåller information om fastigheter och deras ägare. En fastighet är ett mark- eller vattenområde. (I dagligt tal brukar man använda ordet fastighet för byggnad, men det är inte det som avses här.) För fastigheterna ska vi lagra följande information: fastighetsbeteckning, bildningsdatum och taxeringsvärde. Fastighetsbeteckningen består av fyra olika delar: kommun, trakt, block och enhet (ex: Skoby Berga 2:1).

En fastighet består av ett eller flera skiften. Oftast utgörs fastigheten av ett enda skifte (ett sammanhängande markområde), men ibland är fastigheten uppdelad i flera skiften. I en fastighet kan det t.ex. ingå skogsmark som inte ligger i anslutning till den övriga marken. Ett skifte beskrivs av ett skiftesnummer (1, 2, ...), läge angivet i SWEREF 99 TM (ex: N 6097106.67, E 356083.44) samt areal.

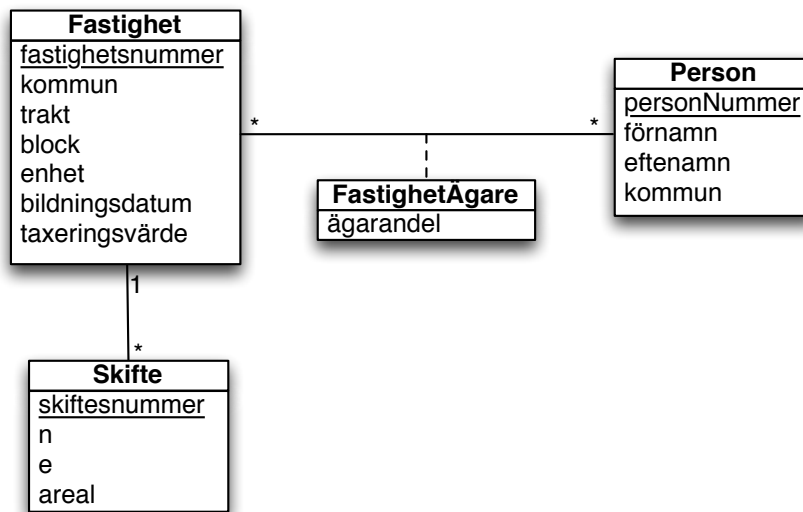
Den information som ska lagras om de personer som äger fastigheter är personnummer, för- och efternamn samt hemkommun. Flera personer kan vara delägare till samma fastighet (anges i %). En person kan äga flera fastigheter.

ER-modell

I beskrivningen av fastighetsdatabasen ovan kan man sluta sig till att vi har följande entitetstyper: Fastighet, Skifte och Person. I fig. 1 visas en ER-modell i UML som beskriver de olika entitetstyperna och sambanden mellan dem.

Fastigheterernas identitet utgörs av fastighetsbeteckningen (kommun, trakt, block, enhet). Men det är opraktiskt att ha en primärnyckel som innehåller fyra attribut. Därför har man lagt till ytterligare ett attribut, fastighetsnummer, som istället kan fungera som primärnyckel.

Hur stor del av en fastighet en viss person äger måste med någonstans i ER-modellen. Attributet ägarandel passar varken hos Fastighet eller hos Person. Istället har man lagt till attributet ägarandel till sambandet mellan Fastighet och Person.



Figur 1: ER-modell

Relationsmodell

ER-modellen kan översättas till följande relationer (tabeller):

Fastigheter(fastighetsnummer, kommun, trakt, block, enhet, bildningsdatum, taxeringsvärde)

Personer(personnummer, förnamn, efternamn, kommun)

Skiften(fastighetsnummer, skiftesnummer, n, e, areal)

FastighetÄgare(fastighetsnummer, personnummer, ägarandel)

Varje entitetstyp motsvaras av en tabell. Lägg dock märke till att även sambanden måste översättas. Tabellerna Skiften och Fastigheter knyts ihop genom att Skifte har fått ett extra attribut - fastighetsnummer. Sambandet mellan Fastigheter och Personer är ett många-till-många-samband och har därför fått en egen tabell.

Öppna fastighetsdatabasen

- Under laborationerna kommer vi att använda databashanteraren Microsoft Access. En fastighetsdatabas med färdiga tabeller finns i filen *fastighetsdatabas_lab.mdb* som hämtas från kursens hemsida:
 - Starta en webbläsare och gå till <http://cs.lth.se/edaa20>.
 - Klicka på Laborationer databaser.

3. Ladda ner filen *fastighetsdatabas_lab.mdb* och spara den på lämpligt ställe.

Klicka på filen *fastighetsdatabas_lab.mdb* så öppnas fastighetsdatabasen i Access.

Access är ett omfattande program där man förutom att skapa tabeller och ställa SQL-frågor även kan skapa frågeformulär och rapporter. Först ska du lära dig använda färdiga tabeller och att skapa frågor. Senare kommer du dessutom att få lära dig hur man skapar tabeller.

I den vänstra panelen syns namnen på de fyra färdiga tabellerna. (För att det tydligt ska framgå vad som är tabeller har vi i den här laborationen valt att låta tabellnamnen börja på tbl, t.ex. tblPersoner. Enligt samma princip låter vi frågornas namn börja med qry.)

Dubbelklicka på ett tabellnamn så öppnas tabellen. Tabellen kan visas i olika vyer. I databladsvyn (eng. Datasheet View) syns den data som tabellen innehåller och här kan man fylla i nya rader med data. När man skapar tabeller använder man designvyn. Man kan växla mellan vyerna i View längst till vänster i verktygslisten. Stäng tabellen.

Skapa frågor

2. Det är dags att skapa SQL-frågor för att hämta information från tabellerna. Det finns två sätt att göra detta:
 - skapa frågan i designvyn.
 - skapa frågan i SQL-vyn.

Skapa en fråga i design-vyn

Skapa en fråga som visar samtliga kolumner i tabellen *tblPersoner* genom att:

1. Klicka på Create i menyraden.
2. Klicka på Query Design.
3. Ett frågefönster och ett popup-fönster visas.
I popup-fönstret kan du välja den eller de tabeller som ska användas i frågan. Välj tabellen *tblPersoner* och lägg till den genom att klicka på Add och stäng sedan popup-fönstret.
4. Frågefönstrets övre halva visar de tabeller som ingår i frågan (här tabellen *tblPersoner*). I den nedre halvan väljer man vilka kolumner (attribut) man vill hämta från de inblandade tabellerna.
Klicka i den första tomma rutan i raden märkt Field och klicka på rullgardinen i det fältet. Välj *tblPersoner.**.
5. Kör frågan genom att klicka på utropstecknet i verktygslisten. Resultatet ska se så här:

personnummer ▾	förnamn ▾	efternamn ▾	kommun ▾
351010	Ola	Setterby	Övered
541111	Olof	Weberup	Skoby
580303	Anna-Karin	Hellström	Skoby
600909	Maria	Svensson	Övered
640101	Nina	Danielsson	Skoby
640808	Maria	Svensson	Lund
651212	Lars	Svensson	Lund
730505	Ulrika	Lifvenborg	Skoby
750202	Adam	Falkenström	Övered
780606	Maria	Lindedahl	Övered
850101	Anita	Nilsson	Lund
900707	Susanne	Nordahl	Skoby

6. Välj spara (klicka på diskettsymbolen längst uppe till vänster) och döp frågan till `qryPersoner`. Stäng frågan.

Skapa en fråga i SQL-vyn

Skapa en fråga som visar samtliga kolumner i tabellen `tblPersoner` genom att:

1. Klicka på Create i menyraden.
2. Klicka på Query Design. Stäng popup-fönstret med tabellerna.
3. Växla till SQL-vyn i View längst till vänster i verktygslisten.
4. Skriv följande SQL-sats:

```
SELECT *
FROM tblPersoner;
```

5. Kör frågan. Resultatet ska bli detsamma som i `qryPersoner`.
6. Stäng frågan. Du behöver inte spara frågan – samma fråga finns ju redan i `qryPersoner`. Öppna gärna `qryPersoner`, växla till SQL-vyn. Notera att frågan ser (ungefär) likadan ut som den fråga du skrev in i SQL-vyn.

Olika vyer av en fråga

Som du redan upptäckt kan man växla mellan olika vyer av frågan:

- designvyn – som används för att skapa frågan genom att välja tabeller, kolumner etc.
- SQL-vyn – som används för att skriva in frågan i SQL.
- databladsvyn – som innehåller resultatet av frågan. Denna vy visas automatiskt när man kör frågan.

Det går alltså att skapa en fråga på två olika sätt; i designvyn respektive i SQL-vyn. Det går också bra att växla mellan de olika sätten. Man kan t.ex. börja med att välja tabeller och kolumner i designläget. Sedan kan man byta till SQL-vyn och fortsätta utveckla frågan.

Även om det ofta går snabbast att skapa frågan i designvyn är det viktigt att du kan formulera frågor i SQL. Det är bra att kunna tyda SQL-koden om en fråga inte ger önskat resultat. Kanske kommer du dessutom i framtiden att använda andra databashanterare eller GIS-programvara där du måste använda SQL.

SQL-frågor

3. I den här uppgiften ska du ställa ett antal frågor till databasen. Skapa frågorna och spara dem under lämpliga namn.

Dina SQL-frågor kan vara bra att ha som exempel om du skulle fortsätta arbeta med databaser. Du som läser på L kommer t.ex. att fortsätta med detta i kursen Geografiska informationssystem. Det kan därför vara en bra idé att fylla i SQL-frågorna i laborationshäftet.

Observera att i frågor framställda i designläget har tabellnamnet lagts till före attributnamnet oavsett om det behövs eller ej (t.ex. `tblPersoner.förnamn` istället för `förnamn`). Att skriva tabellnamnet följt av punkt före attributnamnet behövs egentligen bara om man använder attribut med samma namn från olika tabeller i en fråga.

Tips! Tänk på att du kan växla mellan de olika vyerna av frågan. Det kan vara praktiskt att börja i designläget och välja inblandade tabeller och kanske attribut. Sedan kan man gå över till SQL-vyn för att göra klart frågan. Ibland kan det också vara praktiskt att kopiera en gammal fråga och sedan gå in i SQL-vyn och ändra.

- a) Vilka är personernas namn (förnamn, efternamn)? Sortera efter namnet (i första hand efter efternamn, i andra hand efter förnamn).

SQL:

Resultat:

förnamn ▾	efternamn ▾
Nina	Danielsson
Adam	Falkenström
Anna-Karin	Hellström
Ulrika	Lifvenborg
Maria	Lindedahl
Anita	Nilsson
Susanne	Nordahl
Ola	Setterby
Lars	Svensson
Maria	Svensson
Maria	Svensson
Olof	Weberup

b) Vilka personer kommer från Lund?

SQL:

Resultat:

personnummer ▾	förnamn ▾	efternamn ▾
640808	Maria	Svensson
850101	Anita	Nilsson
651212	Lars	Svensson

c) Vad heter personen med personnummer 651212?

SQL:

Resultat:

förnamn ▾	efternamn ▾
Lars	Svensson

d) Vad har Maria Svensson från Lund för personnummer?

SQL:

Resultat:

personnummer ▾
640808

e) Vilka personer är födda på 70-talet?

SQL:

Resultat:

personnummer	namn	kommun
730505	Ulrika Lifvenborg	Skoby
750202	Adam Falkenström	Övered
780606	Maria Lindedahl	Övered

f) Vilka fastigheter finns? Sortera efter kommun, trakt, block och enhet.

SQL:

Resultat:

fastighetsbeteckning
Skoby Berga 2:1
Skoby Berga 2:2
Skoby Berga 2:3
Skoby Berga 2:4
Skoby Boken 1:1
Skoby Boken 1:2
Skoby Boken 1:3
Skoby Boken 1:4
Skoby Boken 2:3
Skoby Boken 2:4
Övered Berga 2:1
Övered Berga 3:1
Övered Lärbo 1:1
Övered Lärbo 1:2
Övered Lärbo 1:3
Övered Lärbo 1:4
Övered Lärbo 1:7

g) Hur många fastigheter finns det i Skoby kommun?

SQL:

Resultat:

Antal
10

- h) Vilka kommuner förekommer i fastighetsdatabasen? Se till att varje kommunnamn bara visas en gång.

SQL:

Resultat:

kommun ▾
Skoby
Övered

- i) Vilka fastigheter har taxeringsvärde mellan 500 000 och 1000 000 kr? Sortera efter taxeringsvärde.

SQL:

Resultat:

kommun ▾	trakt ▾	block ▾	enhet ▾	taxeringsvärde ▾
Skoby	Berga	2	2	500 000,00 kr
Skoby	Berga	2	1	550 000,00 kr
Skoby	Boken	1	4	875 000,00 kr
Skoby	Boken	1	3	900 000,00 kr
Skoby	Berga	2	4	1 000 000,00 kr

- j) Vilka fastigheter har ett taxeringsvärde som inte ligger i intervallet 500 000 och 1000 000 kr? Sortera efter taxeringsvärde.

SQL:

Resultat:

kommun ▾	trakt ▾	block ▾	enhet ▾	taxeringsvärde ▾
Övered	Lärbo	1	3	300 000,00 kr
Skoby	Boken	2	4	325 000,00 kr
Övered	Berga	2	1	375 000,00 kr
Övered	Lärbo	1	4	400 000,00 kr
Övered	Lärbo	1	2	400 000,00 kr
Övered	Berga	3	1	425 000,00 kr
Övered	Lärbo	1	1	450 000,00 kr
Skoby	Boken	1	2	450 000,00 kr
Skoby	Berga	2	3	1 500 000,00 kr
Skoby	Boken	2	3	1 650 000,00 kr
Övered	Lärbo	1	7	2 300 000,00 kr
Skoby	Boken	1	1	2 400 000,00 kr

k) Vilket är lägsta respektive högsta taxeringsvärdet?

SQL:

Resultat:

Min ▾	Max ▾
300 000,00 kr	2 400 000,00 kr

l) Vilket är medelvärdet av taxeringsvärdena i Övereds kommun?

Tips! I Access kan flera ord som ska utgöra ett namn hållas ihop med [...], ex. [genomsnittligt taxeringsvärde].

SQL:

Resultat:

Genomsnittligt taxeringsvärde ▾
664 285,71 kr

Hämta information från flera tabeller – JOIN

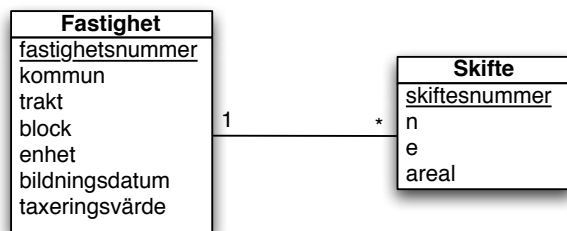
4. Antag att vi vill skapa en lista med fastigheterna och deras skiften. Då behöver vi hämta information från både tblFastigheter och tblSkiften.

Skapa följande fråga i SQL-vyn och provkör.

```
SELECT tblFastigheter.fastighetsnummer, kommun, trakt, block, enhet,
       tblSkiften.fastighetsnummer, n, e, areal, skiftesnummer
FROM tblFastigheter, tblSkiften;
```

Vad blir resultatet? Förklara varför.

Se på tabellen tblSkiften. Notera att fastigheten nr 11 har två skiften. Ett skifte tillhör en och endast en fastighet.



För att skapa kopplingen mellan Fastighet och Skifte använder vi fastighetsnumret. Vi har helt enkelt lagt till primärnyckeln i fastighetstabellen (fastighetsnummer) som en extra kolumn i skiftestabellen. När de bäggefälten är identiska i de bägge tabellerna, då vet vi att fastighet och skifte hänger samman.

Fastighetsnumret är främmande nyckel i skiftestabellen. En främmande nyckel refererar till en primärnyckel i en annan tabell, i det här fallet till fastighetsnummer i fastighetstabellen. Detta innebär att alla fastighetsnummer som förekommer i skiftestabellen måste finnas i fastighetstabellen. Detta kallas referensintegritet.

Rätta till den felaktiga frågan genom att bara ta med de rader där fastighetsnumren stämmer överens. (Använd JOIN). När du fått fram rätt antal rader (20 st) så ta bort de bägge kolumnerna med fastighetsnummer från frågan. Spara frågan under namnet qryFastigheterSkiften.

SQL:

Resultat:

Skoby	Berga	2	3	6546096,72	284626,07	8353144	1
Skoby	Berga	2	2	6548757,20	288410,52	926250	1
Skoby	Berga	2	1	6597106,67	295608,43	529981	1
Skoby	Berga	2	4	6595048,64	282024,06	5443215	1
Skoby	Boken	1	1	6508268,28	293700,26	6754505	1
Skoby	Boken	1	1	7213106,72	281276,71	143250	2
Skoby	Boken	1	1	7212086,38	282653,55	5076	3
Skoby	Boken	1	2	6513201,63	273251,05	248901	1
Skoby	Boken	1	3	6520901,12	273331,55	255453	1
Skoby	Boken	1	4	6526392,70	298323,44	105420	1
Skoby	Boken	2	3	6519636,93	288083,14	4434854	1
Skoby	Boken	2	4	6568593,34	289270,20	14890	1
Övered	Lärbo	1	7	6589134,04	807833,40	5421050	1
Övered	Lärbo	1	7	7212392,62	805398,45	53075	2
Övered	Lärbo	1	3	6509293,75	814467,12	677221	1
Övered	Lärbo	1	2	7225449,11	821298,14	523450	1
Övered	Lärbo	1	1	7232168,17	799639,19	523680	1
Övered	Lärbo	1	4	7236548,53	819498,74	322080	1
Övered	Berga	2	1	7259745,67	820202,15	122120	1
Övered	Berga	3	1	7266089,69	818716,55	223890	1

Markera samband mellan tabeller – stöd för JOIN

5. Så länge vi jobbar i SQL är det vi själva som håller reda på hur vi har kopplat tabeller med varandra i vår tabelldesign. Dock har Access stöd för att definiera samband mellan tabeller som del av databasen, vilket gör att Access också kan skapa JOIN-operationer automatiskt. Om vi definierar sambanden i Access får vi även grafiskt stöd, vilket är en stor hjälp.

Man kan definiera samband i ett verktyg i Access som heter Relationships. Starta sambandsverktyget genom att klicka på Database Tools i menyraden och därefter klicka på Relationships. I det fönster som dyker upp visas tabeller som grafiska objekt och samband mellan dem.

Fokusera på tabellerna tblFastigheter och tblSkiften. Dra fastighetsnumret från det ena tabellen och släpp det på fastighetsnumret i den andra ("Drag-and-drop"). I pop-up-fönstret klickar du i rutan för referensintegritet (Enforce referential integrity) och sedan klickar du på Create. Resultatet ska bli en linje mellan attributen, en indikation på att attributen är sammankopplade.

Access "förstår" att det endast kan finnas en fastighet kopplad till varje skifte eftersom fastighetsnumret i fastighetstabellen är definierad så att det inte får finnas några dubletter (definitionen på en primärnyckel). Access "förstår" på samma sätt att det kan finnas flera skiften till varje fastighet eftersom fastighetsnumret i skiftestabellen är definierad så att det får finnas dubletter.

Observera att det du ser i sambandsverktyget inte är ER-modellen utan en bild av de färdiga tabellerna.

Stäng sambandsverktyget. Skapa en fråga i designläge. Lägg till tblFastigheter och tblSkiften i pop-up-fönstret och stäng sedan pop-up-fönstret. Man ser hur Access grafiskt visar att det finns ett samband mellan tabellerna.

Välj kolumnerna kommun, trakt, block, enhet, n, e, areal och skiftesnummer.

Kör frågan. Resultatet ska bli detsamma som i qryFastigheterSkifte.

- c) Vilka fastigheter (kommun, trakt, block, enhet, ägarandel) äger Lars Svensson?
SQL:

Resultat:

kommun ▾	trakt ▾	block ▾	enhet ▾	ägarandel ▾
Skoby	Boken	1	4	100,00%
Övered	Lärbo	1	7	50,00%

- d) Skapa en fastighetsförteckning, en lista som innehåller fastighetsbeteckning, namn samt ägarandel. Sortera efter fastighetsbeteckningen.
SQL:

Resultat:

fastighet ▾	ägare ▾	ägarandel ▾
Skoby Berga 2:1	Lindedahl, Maria	100,00%
Skoby Berga 2:2	Hellström, Anna-Karin	100,00%
Skoby Berga 2:3	Lifvenborg, Ulrika	100,00%
Skoby Berga 2:4	Danielsson, Nina	100,00%
Skoby Boken 1:1	Svensson, Maria	100,00%
Skoby Boken 1:2	Svensson, Maria	100,00%
Skoby Boken 1:3	Weberup, Olof	100,00%
Skoby Boken 1:4	Svensson, Lars	100,00%
Skoby Boken 2:3	Nordahl, Susanne	100,00%
Skoby Boken 2:4	Danielsson, Nina	100,00%
Övered Berga 2:1	Svensson, Maria	100,00%
Övered Berga 3:1	Svensson, Maria	100,00%
Övered Lärbo 1:1	Falkenström, Adam	60,00%
Övered Lärbo 1:1	Setterby, Ola	40,00%
Övered Lärbo 1:2	Hellström, Anna-Karin	100,00%
Övered Lärbo 1:3	Falkenström, Adam	100,00%
Övered Lärbo 1:4	Nordahl, Susanne	100,00%
Övered Lärbo 1:7	Svensson, Maria	50,00%
Övered Lärbo 1:7	Svensson, Lars	50,00%
Övered Lärbo 1:7	Lindedahl, Maria	100,00%

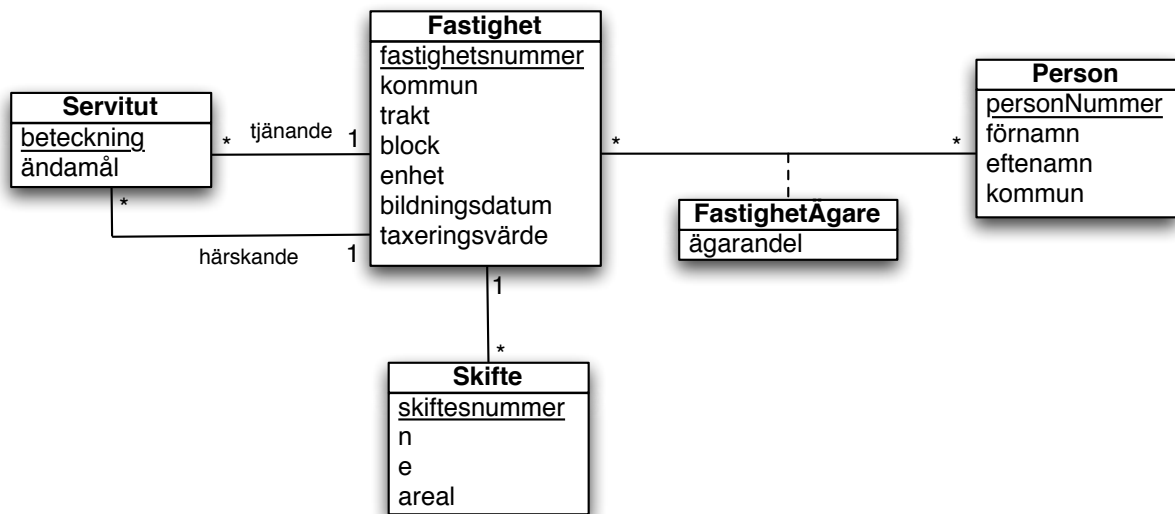
Servitut

8. Läs igenom texten om servitut nedan och studera ER-modellen och de tabeller som blir resultatet när man översätter ER-modellen till relationsmodell. Diskutera ev. oklarheter med labhandledaren.

Servitut innebär rätt för en fastighet att använda en annan fastighet på ett visst sätt. Rätten kan avse utfart, tillgång till brunn och liknande. Den fastighet som har nytta av servitutet kallas den härskande fastigheten medan den fastighet på vilken servitutet lagts kallas den tjänande fastigheten.

I den här uppgiften förutsätter vi att ett servitut bara rör två fastigheter; den härskande och den tjänande. Se fig. 2.

Ett servitut har en beteckning (ex: 2150-86/322.2) och ett ändamål (ex: väg).



Figur 2: ER-modell

Databasen kommer nu att innehålla följande relationer (tabeller):

Fastigheter(fastighetsnummer, kommun, trakt, block, enhet, bildningsdatum, taxeringsvärde)

Personer(personnummer, förnamn, efternamn, kommun)

Skiften(fastighetsnummer, skiftesnummer, n, e, areal)

FastighetÄgare(fastighetsnummer, personnummer, ägarandel)

Servitut(beteckning, ändamål, härskande, tjänande)

Skapa tabell

9. Skapa tabellen Servitut genom att:
1. Klicka på Create i menyraden.
 2. Klicka på Table Design. Ett fönster där du kan fylla i attributnamn och attributens typ visas.
 3. Första attributet ska vara en sträng med namnet beteckning. Klicka i den tomma rutan under Field Name och fyll i namnet beteckning. Klicka sedan i rutan under Data Type. Genom att klicka på rullgardinen kan man välja attributets typ. Använd

den förvalda typen Short Text. Teckensträngens längd (Field Size) har det förvalda värdet 255. Ändra till 20 (20 tecken räcker).

Attributet beteckning ska fungera som primärnyckel. Ange detta genom att först klicka i rutan med beteckning och sedan klicka på Primary key i verktygslisten.

4. Skapa på liknande sätt de övriga attributen. Börja med attributet ändamål som ska ha typen Short Text. Låt teckensträngens längd ha det förvalda värdet 255.
5. Skapa attributen härskande och tjänande med typen Number. Låt rutan Field Size ha det förvalda värdet Long Integer.

Välj Yes i rutan Required. Det innebär att man inte kan utelämna uppgift om härskande och tjänande fastighet i raderna med data.

Vi vill att Access snabbt ska kunna söka på härskande respektive tjänande fastighet. Detta åstadkommer man genom att låta dessa attribut vara indexerade. Välj värdet Yes (Duplicates OK) i rutan Indexed.

Att avgöra vilka attribut som ska vara indexerade är inte helt enkelt. Genom att låta ett attribut vara indexerat går det snabbare att söka på det. Men uppdateringar av tabellen (sätta in nya rader, ändra ett värde på en rad etc.) tar längre tid.

6. Spara tabellen under namnet tblServitut och stäng tabellen.

Starta sambandsverktyget (Relationships i menyn Database Tools). Högerklicka i fönstret och välj Show Tables. Lägg till tabellen tblServitut. Skapa de två sambanden mellan tabellerna tblFastigheter och tblServitut. Glöm inte att kryssa i rutan för referensintegritet. (Eftersom det är samma attribut i tabellen tblFastigheter som är inblandade i de bägge sambanden visas två bilder av den tabellen.) Stäng sambandsverktyget.

Kul att veta: Tabeller kan också skapas genom en SQL-sats. Så här ser frågan som skapar tabellen tblServitut ut:

```
CREATE TABLE tblServitut (
  beteckning VARCHAR(20) PRIMARY KEY,
  ändamål VARCHAR(255),
  härskande INT REFERENCES tblFastigheter(fastighetsnummer) NOT NULL,
  tjänande INT REFERENCES tblFastigheter(fastighetsnummer) NOT NULL
);
```

10. Nästa steg i processen är att fylla tabellen med data. I verkligheten hade vi – eller någon annan – skapat ett mer användbart formulär, kanske med webbgränssnitt, för att fylla på uppgifter om servitut. I gränssnittet mot användaren hade det inte synts att det finns flera tabeller inblandade. Istället hade användaren fyllt i uppgifter i ett för ändamålet utformat gränssnitt. Översättningen från formulär till rätt tabeller blir en fråga för programmeraren.

Om man har sin data i t.ex. Excel eller i en semikolonseparerad textfil kan man importera data till tabellen. Det finns olika format för import och proceduren är ganska enkel.

På den här laborationen ska du istället skriva in informationen rakt in i tabellen.

Öppna tabellen tblServitut och välj databladsläge. En tom tabell syns med de definierade kolumnerna från förra deluppgiften. Fyll fälten med data enligt tabellen nedan:

beteckning ▾	ändamål ▾	härskande ▾	tjänande ▾
2150-86/322.1	Väg	2	4
2150-86/322.2	Väg	3	4
2160-90/10.1	Ledning	13	14
2160-90/10.2	Brunn	13	14

Prova att fylla i ett värde på tjänande eller härskande fastighet som inte finns i fastighetstabellen.

Vad händer? Förklara varför.

SQL-frågor

11. a) Vilka fastigheter belastas av servitut?
SQL:

Resultat:

tjänande	kommun	trakt	block	enhet	beteckning	ändamål	härskande
4	Skoby	Berga	2	4	2150-86/322.1	Väg	2
4	Skoby	Berga	2	4	2150-86/322.2	Väg	3
14	Övered	Lärbo	1	1	2160-90/10.1	Ledning	13
14	Övered	Lärbo	1	1	2160-90/10.2	Brunn	13

Tips! Om du skapar frågan genom att välja de båda inblandade tabellerna i designvyn så får du förmodligen en tom tabell som resultat. Om man ser på frågan i SQL-vyn kan man ana varför. Problemet är att det finns fler samband i tabellerna än de vi vill titta på i den här specifika frågan. Access försöker med sin automatik att ta med samtliga samband, och går på det sättet vilse i sina JOIN-operationer.

Det finns två sätt att lösa problemet. Det ena är att helt jobba i SQL och strunta i designvyn. Det andra sättet är att helt enkelt ta bort den eller de samband vi inte är intresserade av i frågan. Att ta bort sambandslinjerna i frågeverktyget påverkar inte de underliggande sambanden i Access som vi har definierat i sambandsverktyget.

Ta bort sambandet mellan fastighetsnummer i fastighetstabellen och härskande i servitutstabellen (högerklicka på linjen och välj Delete).

- b) Vilka servitut finns till förmån för fastigheten Skoby Berga 2:2?
SQL:

Resultat:

beteckning	ändamål	tjänande
2150-86/322.1	Väg	4

Parametrar

12. I Access finns möjlighet att använda parametrar istället för att i frågans WHERE-del specificera vilka rader man är intresserad av.

Kopiera sista frågan från föregående uppgift (som visar vilka servitut som finns till förmån för fastigheten Skoby Berga 2:2) och spara den under ett nytt namn. Byt ut Skoby mot [Kommun?] genom att ändra direkt i frågan i SQL-läget eller i villkorsfältet i designläget. Byt på motsvarande sätt ut Berga, 2 och 2.

Kör frågan och kontrollera att den ger rätt resultat för några olika fastigheter.

Här används Kommun? som parameter-namn. Du får gärna välja något annat namn på parametern (utom attributnamnen). Frågetecknet behövs ej, utan är bara med för att det ska se snyggare ut i popup-rutan. Man kan skriva utförligare text, t.ex. [Ange önskad kommun].

Gruppering

13. Ibland vill man gruppera raderna i en tabell och behandla varje grupp för sig. I den här databasen kan vi lätt beräkna medelvärdet av taxeringsvärdet för samtliga fastigheter genom att ställa frågan

```
SELECT AVG(taxeringsvärde) AS medelvärde  
FROM tblFastigheter;
```

För att få fram medelvärdet av samtliga fastigheters taxeringsvärden per kommun måste man gruppera efter kommuner (GROUP BY). Ställ frågor som visar följande:

- a) Visa medelvärdet av samtliga fastigheters taxeringsvärden i de olika kommunerna.
SQL:

Resultat:

kommun ▾	medelvärde ▾
Skoby	1 015 000,00 kr
Övered	664 285,71 kr

b) Visa totalarealen för varje fastighet.

SQL:

Resultat:

kommun ▾	trakt ▾	block ▾	enhet ▾	totalareal ▾
Skoby	Berga	2	1	529981
Skoby	Berga	2	2	926250
Skoby	Berga	2	3	8353144
Skoby	Berga	2	4	5443215
Skoby	Boken	1	1	6902831
Skoby	Boken	1	2	248901
Skoby	Boken	1	3	255453
Skoby	Boken	1	4	105420
Skoby	Boken	2	3	4434854
Skoby	Boken	2	4	14890
Övered	Berga	2	1	122120
Övered	Berga	3	1	223890
Övered	Lärbo	1	1	523680
Övered	Lärbo	1	2	523450
Övered	Lärbo	1	3	677221
Övered	Lärbo	1	4	322080
Övered	Lärbo	1	7	5474125

c) Om man vill välja ut vissa grupper (rader) när man grupperat kan man använda HAVING. Ställ samma fråga som ovan men tag bara med fastigheter som har en areal på minst 500000 kvm.

SQL:

Resultat:

kommun ▾	trakt ▾	block ▾	enhet ▾	totalareal ▾
Skoby	Berga	2	1	529981
Skoby	Berga	2	2	926250
Skoby	Berga	2	3	8353144
Skoby	Berga	2	4	5443215
Skoby	Boken	1	1	6902831
Skoby	Boken	2	3	4434854
Övered	Lärbo	1	1	523680
Övered	Lärbo	1	2	523450
Övered	Lärbo	1	3	677221
Övered	Lärbo	1	7	5474125

d) Visa totalarealen för fastigheten Övered Lärbo 1:7.

Tips: Utgå från frågan i deluppgift b.

SQL:

Resultat:

kommun ▾	trakt ▾	block ▾	enhet ▾	totalareal ▾
Övered	Lärbo	1	7	5474125

Man kan också gruppera i designläget. Man börjar som vanligt med att välja tabeller och därefter de kolumner som ska vara med (kommun, trakt, block, enhet, areal).

Därefter är det dags att gruppera. Klicka på Totals i verktygslisten. I nedre delen av fönstret där man anger kolumner, villkor m.m., visas då en rad som heter Total. Den förvalda funktionen är Group By. Nu grupperas det på samtliga attribut, d.v.s. om, och endast om, samtliga attribut vi har angett är lika så slår vi samman raderna till en rad. Det vi vill göra är att slå samman alla rader där fastighetsbeteckningen är identisk. Alltså ska vi inte ha Group By i fältet för Areal. Klicka i Total-fältet för Areal och välj i stället Sum. Den funktionen innebär att vi summerar för varje rad vi slår samman med grupperingsvillkoret.

Subfrågor

Ibland behöver man använda resultatet från en annan fråga (subfråga) inuti ett villkor. Ställ följande frågor:

14. a) Vilken fastighet har det lägsta taxeringsvärdet?

SQL:

Resultat:

kommun ▾	trakt ▾	block ▾	enhet ▾	taxeringsvärde ▾
Övered	Lärbo	1	3	300 000,00 kr

- b) Vilka fastigheter belastas *inte* av servitut?

SQL:

Resultat:

fastighetsnummer ▾	kommun ▾	trakt ▾	block ▾	enhet ▾
1	Skoby	Berga	2	3
2	Skoby	Berga	2	2
3	Skoby	Berga	2	1
5	Skoby	Boken	1	1
6	Skoby	Boken	1	2
7	Skoby	Boken	1	3
8	Skoby	Boken	1	4
9	Skoby	Boken	2	3
10	Skoby	Boken	2	4
11	Övered	Lärbo	1	7
12	Övered	Lärbo	1	3
13	Övered	Lärbo	1	2
15	Övered	Lärbo	1	4
16	Övered	Berga	2	1
17	Övered	Berga	3	1

Använda frågeresultat som tabell

15. Istället för att använda subfrågor eller när man har ännu mer komplicerade frågor kan man använda resultatet från en fråga som om den vore en tabell.

Vi ska ta reda på fastighetsbeteckning och totalareal för den fastighet som har minst areal.

Skapa en ny fråga i design-läge. Istället för att lägga till en eller flera tabeller så lägg istället till *frågan* som listar fastigheternas totalareal (uppgift 14 b). Ta med kolumnerna kommun, trakt, block, enhet och totalareal. Gå till SQL-vyn. Ändra frågan så att bara den fastighet som har minsta totalarealen visas. Tips. Lägg i WHERE-delen till villkoret att totalarealen ska vara lika med den minimala totalarealen. Den minimala totalarealen får man fram genom en subfråga.

SQL:

Resultat:

kommun ▾	trakt ▾	block ▾	enhet ▾	totalareal ▾
Skoby	Boken	2	4	14890

Laboration 2 – UML

På den här laborationen ska du träna på att gå från problem till databas. Du ska först göra en ER-modell av problemet. E/R står för entity/relationship och ska uttrycka vilka entitetstyper ("saker") som finns i problemet och hur de hänger ihop. ER-modellen ska uttryckas i UML. Sedan ska du översätta ER-modellen till en relationsmodell. Relationsmodellen visar vilka tabeller som ska finnas.

Mål

Exempel på vad du ska kunna efter laborationen:

- Rita en ER-modell i UML (med entitetstyper, attribut, samband, multiplicitet).
- Översätta en ER-modell till en relationsmodell.
- Förstå hur man översätter ett-till-många-samband med ett extra attribut i en tabell och hur man översätter många-till-många-samband med en extra tabell.
- Förklara begreppen primärnyckel, främmande nyckel, referensintegritet.

Förberedelser

- Läs bilderna från andra databasföreläsningen.
- Läs kundens beskrivning av problemet nedan.
- Börja skissa på ER-modellen.

Kundens beskrivning

Blomstermåla Rockfestival går av stapeln torsdag till lördag, med band från hela världen som uppträder från tidiga eftermiddagen till sena natten. Festivalen har nu hållit på i snart tio år, och har vuxit så pass att den behöver ett datasystem för att hålla reda på alla som jobbar, spelscheman, band och bandmedlemmar och liknande information. Mycket av arbetet för att hålla reda på alla jobbares person- och arbetsuppgifter, sammanställa programmet och liknande har hittills gjorts för hand, och kanslipersonalen ser med stor tillförsikt fram emot den hjälp de kommer att få av ert system. Här nedan följer några beskrivningar av uppgifter som vi hoppas att datorsystemet ska klara av.

Spelscheman

Den huvudsakliga informationen som allmänheten vill ha från kansliet är ju spelschemana för de olika banden: Vilket band som spelar på vilken scen vid vilken tidpunkt. Festivalen har ett flertal scener, både små för en publik på ca 50 personer till stora scenen med plats för 500. För det mesta spelar ett band bara en gång under festivalen, men det händer att särskilt populära band får flera scentider. Att hålla reda på när de olika banden ska gå på och av de olika scenerna är mycket tidsödande om man försöker göra det med kylskåpsmagneter och papperslappar, så vi hoppas mycket på datorns hjälp just här! Praktiska listor för varje scen med bandnamn, tider, ursprungsland och kanske musikstil vore fantastiskt bra att ha.

Säkerhetsansvar

Vi har ett krav på oss från brandmyndigheterna att vid varje tidpunkt ha en person som är säkerhetsansvarig för varje scen. Det löser vi som så att vi ger särskilt ansvarsställande och rutinerade festivaljobbare ansvar för de olika scenerna i fyratimmarspass under kvällarna. Vanligtvis är det ett lugnt arbete, men om det skulle inträffa en incident med till exempel pyrotekniken så är det viktigt att det finns någon som vet vem man ringer och hur man löser situationerna effektivt.

Både för vår egen skull och för att brandmyndigheterna kräver det, vill vi ha möjligheter att få ut listor på vilka som har säkerhetsansvaret för de olika scenerna olika kvällar. Listorna måste innehålla jobbarnas personnummer, eftersom brandchefen måste kunna kontrollera åldern på dem.

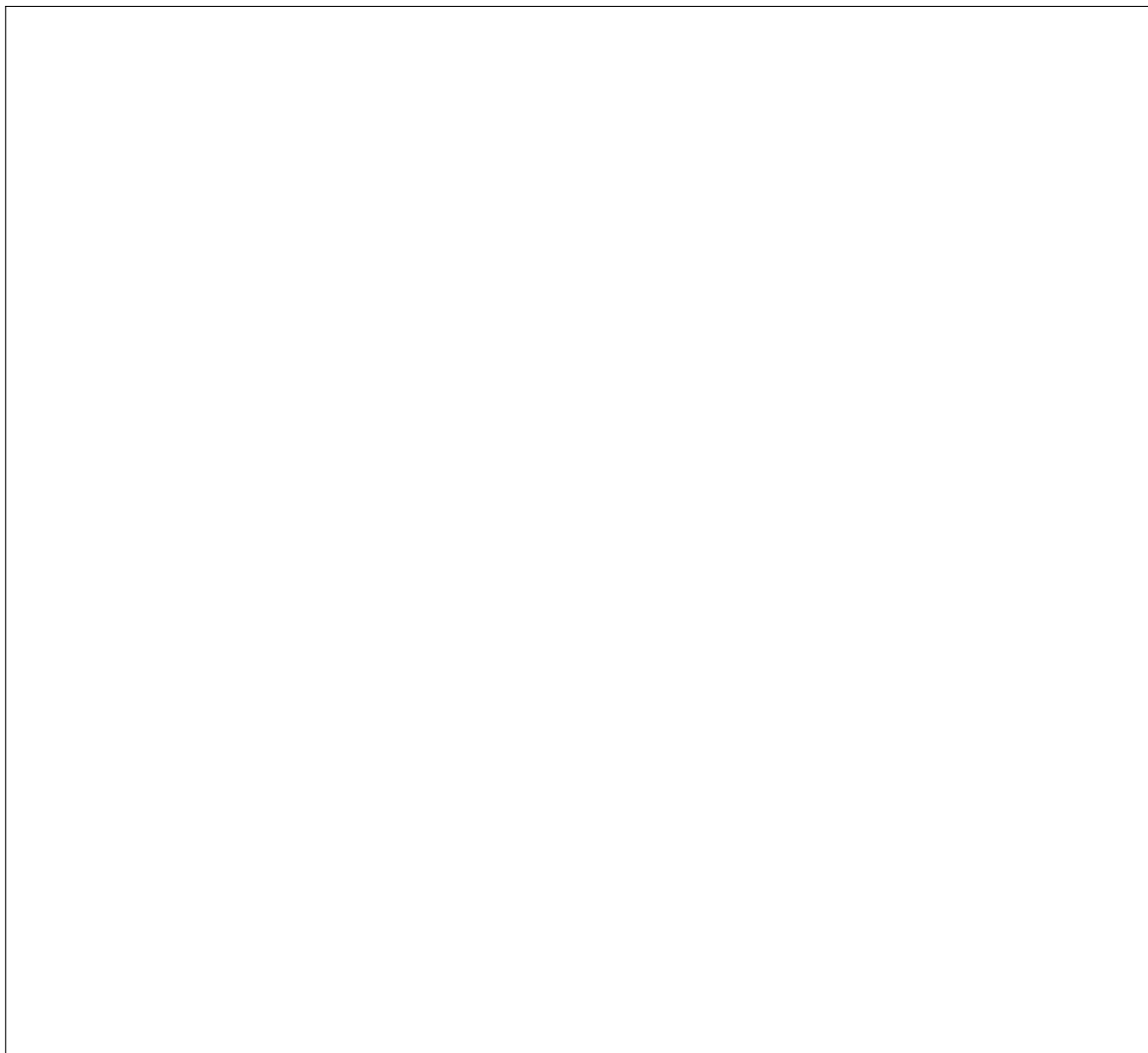
Kontaktpersoner för banden

Varje band som spelar på festivalen får sig tilldelad en kontaktperson bland jobbarna. Denna person är ansvarig för att hjälpa sittband till loger och scener, lösa småproblem som kan dyka upp, osv. Samtidigt är de festivalledningens kanal till banden om det är någon information som behöver komma ut snabbt. En person kan ha ansvar för flera band. Idealt så skulle kansliet vilja kunna få fram både namnet på kontaktpersonen för ett givet band, en lista över alla bands kontaktpersoner och en lista över alla jobbare som är kontaktpersoner och vilka band de är ansvariga för.

Så, det var lite tankar från kansliet om vad vi skulle vilja ha hjälp med. Vi hoppas mycket på er och ert databassystem!

Uppgifter

1. Utveckla en ER-modell med entitetstyper, attribut, samband som beskriver problemet ovan. Ange multipliciteter och markera primärnycklar. Tänk på att använda tydliga namn på entitetstyper och attribut.



2. Utveckla en relationsmodell som visar vilka tabeller som blir resultatet av ER-diagrammet från förra uppgiften? Beskriv tabellerna på formen Tabell(attribut1, attribut2, ...). Stryk under primärnycklarna.



3. Ge några exempel på en främmande nyckel och ange också vad den refererar till (tabell och attribut):

