# Stickalster – bakgrund till enskild inlämningsuppgift i databasteknik

*Anna-Maja Lithner*

Sedan 2014 har jag stickat mycket – på beställning eller på eget initiativ som gåva. Länge kunde jag minnas vilka alster jag producerat, men har också tänkt att någon gång skriva upp dem. Påfallande ofta har jag också tänkt, inför ett nytt alster, att det varit bra att kunnat gå tillbaka till tidigare stickningar för att få idéer om materialval.

Det har därför fallit sig naturligt att mina stickade alster är det som står i fokus för slutuppgiften i databasteknik. Dels för att det är ett trevligt sätt att dra mig till minnes de projekt jag jobbat med, men också för att jag kan se att jag faktiskt kommer att kunna ha nytta av systemet även sedan.

Processen innan jag har ett alster färdigt kan se olika ut.

Ibland har jag ett garn hemma som jag är sugen på att skapa något av. Ibland har jag en längtan efter att tillverka en speciell sak. Ibland har jag en önskan om att få sticka något till just en specifik person. Ibland får jag beställningar på något specifikt, som någon vill ha till sig själv eller till någon närstående. Några gånger har jag stickat utan att ha någon given mottagare, och sålt i efterhand.

Jag köper mina garner från olika håll, ibland använder jag mig av garn som är återvunnet från gamla plagg.

Det jag har tänkt är intressant att lagra, är först och främst att samla ihop alla alster som blivit sedan jag började. Men i den processen blir det också relevant att samla information om de olika material som använts, och samla alla mottagare på ett och samma ställe.

Jag har inte inhämtat samtycke till att lagra uppgifter från mer än någon enstaka person. Jag är därför mån om att ha minimal information och kommer därför bara att lagra namn enligt ”minsta möjliga information”.

Min förhoppning med databasen, är att det skall vara enkelt att lägga in ny information om avslutade alster, samt att plocka fram information om specifika alster (materialval, stick-tjocklek, teknik), och sammanfattning av vad olika personer har som jag tillverkat.

Eftersom all hittillsvarande information läggs in i efterhand, kommer per automatik många fält att behöva lämnas tomma. Jag kommer inte att kunna rekonstruera vilken dimension av stickor som använts, eller hur mycket jag fått i ersättning. Alla fält jag har med, kommer jag med största sannolikhet dock att ha användning för.

## Sammanfattning av projektet vid inlämning

Jag skulle ha behövt mycket mer hjälp med att designa modellen, framförallt i relationsmodell-fasen. Inser att det finns mycket stora luckor i hur de olika tabellerna länkas, samt med vilka värden som skall fyllas i. Alla script är körbara, men resultatet är otroligt svåranvänt.

## Det jag lärt mig under arbetet

sammantaget med arbetet med gruppinlämningsuppgiften:

* Skapa ER-diagram
* Omvandling från ER-diagram till relationsmodell
* Omvandling från modell till tabell. Plötsligt känns de första tre stegen ganska begripliga.
* Återskapa droppat data med befintliga script – learning by making major mistakes.
* Skapa vyer, skapa procedurer
* Joina, selecta – fått en fantastiskt bra steg-för-steg-beskrivning (du ska få den för den skulle kunna hjälpa nästa förvirrade klass)
* Det är lätt att blanda ihop vad som är en vy == vilka data jag ser samtidigt, och vad som är en tabell == vilka data som lagras ihop.
* Fortsatt arbete med github, som påbörjades under OOP. Har haft konkret nytta av att kunna gå tillbaka till tidigare körbara versioner, eftersom systemet vid några tillfällen helt oförklarligt har buggat bort grejer. Enklare nivå på github än under OOP, dock – har inte haft några branches etc, men pushat åtm en gång dagligen. Praktiskt med github under gemensamma uppgiften också, eftersom jag och min parhäst fick lite smått om gemensam tid på slutet == jag vet att hon lätt skulle kunna öppna all kod därifrån och lagra hos sig, och hon har redan förberett så alla schemanamn etc är desamma som jag hade.

Mer dyrköpta lärdomar är vikten av att ha en klar bild av hur systemet skall se ut och fungera på modellstadiet, samt vikten av att lagra kod på ett sätt som gör det enkelt att droppa och göra om, då ändringar krävs. Det system jag hade skapat, för att organisera arbetet, gjorde den processen enormt tidsödande. Gjorde om det systemet under arbetet med enskilda uppgiften.

Jag tycker att jag har behövt lägga väldigt mycket fokus på lite fel saker under kursen, vilket gjort att det varit svårt att hinna riktigt få en förståelse för t ex triggers. Återkommer till det i utvärdering / förbättringsförslag.

Jag känner inte att jag ännu har något grepp om hur triggers konstrueras. Jag förstår dock:

* Hur de fungerar – när jag utför något med en enhet, skall automatiskt något annat ske.
* Kan se många fördelar med att automatiskt kunna generera något nytt i samband med att en förändring görs. Minskar risken för dataförlust, redundans, kontaminerade tabeller etc.
* Principen med cascade – att om data ändras någonstans, skall samma ändring i motsvarande data göras överallt där det förekommer.

**Exempel på konstaterade problem med modellen:**

Jag ser att det blir opraktiskt med att bara ha id siffra som fk, när jag skall lägga in värden i länkade tabeller. Får fundera på om det är så himla smart. Att ha fler poster som kombinerad fk blir å andra sidan lite omständigt, och frågan är om det ändå löser problemet.

Enligt instruktionerna:

Grönt – klart

Rött – inte klart

* **KLART** Välj en egen databas med ett syfte som intresserar dig
* **KLART** ER-diagram över de olika entiteterna som du ser finns i systemet samt rita ut hur de relaterar till varandra.
* **KLART** Utifrån ER diagrammet, skapa en relationsmodell (crow-foot) där du beskriver samtliga delar i databasen, dvs tabeller (minst 5), kolumner, datatyper, relationer, eventuell referensintegritet och index etc. Databasmodellen skall vara av 3NF i så stor utsträckning som möjligt, de-normalisering skall motiveras.
* **KLART** Skapa sedan databasschemat med DDL script samt lägg in viss grunddata om det behövs.
* DU skall sedan skapa DML-script för att kunna hantera relevanta frågor (motsvarnde uppgift G1) samt minst 2 lagrade procedurer, 1 vy och 1 trigger.

**Leverabler *allt zippat till en fil***

* Ingen dagbok behövs **KLART** - har ändå gjorts, analogt med gruppinlämningen.
* ER-diagram (bild) **KLART – är inte aktuell i förhållande till rel-modell och DB**
* Relationsmodell (.mwb) **KLART**
* DDL-script för att skapa databasen (.sql) **KLART – bör vara korrekt version**
* DML-script för att skapa initialt data (.sql) **KLART – bör vara korrekt version, har uppdaterats allt eftersom**
* DML-script för de olika frågorna (.sql) som inte är procedurer
* En enkel beskrivning av vad du gjort, hur du gått tillväga, vad som varit utmanande samt hur många timmar du lagt ner (max två A4). Antalet timmar du lagt ner är ENDAST till för min planering för framtiden, inte för att bedöma resultatet! **KLART – vet inte om det är en enkel beskrivning …?**