

Torsdag 17 nov – ~~Installera SQL Server~~ Datamodellering

Mindre ändring i schemat: Installera SQL Server avhandlas måndag 21 november istället.

DBeaver

En sak kan vi dock smygstarta med redan idag:

Installera DBeaver Community, ett klientprogram för att hantera databaser av olika slag.

<https://dbeaver.io/download/>

Några korta ord om transaktioner

Användbart vid test av olika SQL-kommandon:

- Påbörja transaktion med “begin transaction”,
- ångra kommandon med “rollback transaction”,
- eller bekräfta alla gjorda ändringar med “end transaction” (eller “commit”)

Mer om transaktioner kommer nästa vecka.

Modellera data, Skapa databaser

Objekt och deras egenskaper.

Jämför med klasser och attribut inom objektorienterad programmering.

Alternativa lösningar vid utformning av en databas.

Vad är den största gemensamma nämnaren för en grupp av objekt?

Kan vi tänka oss något slags arv eller gradvis specialisering?

Relationer av olika typer: En till många, Många till en, En till en, Många till många.

ER-diagram.

Kardinalitetsförhållanden.

Olika sätt att notera relationer mellan objekt eller entiteter.

Inte bara entiteter och relationer, utan också affärsregler mm.

Viktigt att stämma av med flera olika intressenter.

Kan vissa krav komma att ändras i framtiden?

Även det man har valt bort kan vara bra att få dokumenterat.

Normalisera databaser, med olika grader av strikthet

Olika aspekter att beakta:

- Redundanta data, som har både för- och nackdelar
- Både svårare och enklare att uppdatera och hämta data
- Inkonsistens / Inkonsekvens

Steg på vägen mot normalisering:

- “Var sak på sin plats”, Skapa en tabell per uppsättning data som hör ihop
- Placera data som upprepas i separata tabeller
- Dela upp sammansatta egenskaper på minsta möjliga beståndsdelar (atomära värden)
- Identifiera varje post med en unik nyckel
- Länka ihop med främmande nycklar

Kan man gå för långt? Svar ja.

Vilka för- och nackdelar ska vägas mot varandra?

När är hårdkodade värden bättre (eller i alla fall helt OK) jämfört med dynamiska värden i uppslagstabell?

Behöver man kunna allt om första, andra, tredje, fjärde och femte normalformerna? Svar nej.

Viktigare är att kunna göra rimliga avvägningar, och att kunna motivera eventuella undantag och avvikelser.

Nycklar, primära och främmande

En primär nyckel är det värde (eller kombination av värden) som unikt identifierar en rad i en tabell.

Vad som är “tillräckligt unikt” kan variera mellan olika tillämpningar och miljöer.

En bekväm lösning är ett numeriskt id som räknas upp automatiskt.

Där samma värde förekommer i en annan tabell är det en främmande nyckel.

Begreppet “referentiell integritet”.

Vi vill inte skapa föräldralösa rader, som kan ge oväntade resultat i vår applikation.

För- och nackdelar med “cascade update” och “cascade delete”?

Var ska kontroller ske?

- I databas.
- I applikation.
- På båda ställena.

Översätta relationer mellan entiteter till tabeller i databas

Vägen från konceptuell eller logisk modell till fysisk modell.

Den logiska modellen bör vara plattformsoberoende.

Den fysiska modellen bör innehålla mer av systemspecifika egenskaper.

Hur översätta många-till-många-relationer till tabeller i databas?

Övningar

Gör klart Exercises på <https://www.w3schools.com/sql/exercise.asp> (nu även “order by”, “functions”, “join”, “group by” och “database”)

Eller motsvarande övningar på <https://sqlbolt.com/> eller liknande platser.

Utifrån exemplet med Students och Courses, fundera på lösningar för att ...

- Upprätthålla regler för obligatoriska värden, för att undvika dubletter, osv.
- Lägga till regler för de kolumner som utgör främmande nycklar i kopplingstabellen StudentsCourses.
- Ändra kolumnen grade till en annan datatyp än char(3) och placera de betyg som kan förekomma i en separat tabell.
- Göra urval som visar vilka kurser de olika studenterna är registrerade på.
- Göra urval som visar hur många studenter som är registrerade på respektive kurs.

Använd kommandotolk, DB Browser for Sqlite, Dbeaver, eller valfritt annat verktyg.

Bonusuppgifter:

- Om vi skulle lägga till en tabell Persons som hanterar personer, hur skulle vi behöva ändra tabellen Students till att hantera endast det som är kopplat till att vissa personer är studenter?
- Kan vi använda Persons även för att peka ut vem som är lärare på respektive kurs?
- Om en kurs har två eller flera lärare, hur gör vi då?

Öva gärna på att rita diagram med tex <https://app.diagrams.net/> .