

ER DIAGRAMÖVNINGAR

Tips:

1

Om det inte finns någon unik nyckel i en entitetstyp, så kan man skapa ett särskilt unikt identitetsnummer bara för användning i databasen.

2

Det är ofta klokt att skapa ett unikt identitetsnummer även om det redan finns en unik nyckel, om den nyckeln är en textsträng eller olämplig av andra skäl.

3

Man behöver inte använda arv eller svaga entitetstyper i dom här uppgifterna.

Kom ihåg:

1

Finns det saker i rutan ovan som är svåra eller omöjliga att rita i ER-diagrammet? Hur gör man då?

2

Fattas det information som egentligen skulle behövas för att rita ER-diagrammet? Hur gör man då?

Gör det här:

Skapa ett entity-relationship-schema för att skapa databasen i varje uppgift, ni behöver inte ange datatypen (domänen) för varje attribut, men ni bör ange primär- och främmande- nycklar.

Övning 1: Gamingsoft: Multi User Dungeon Game

Det finns en klass av dataspel som kallas **massively multiplayer online role-playing games**, eller **MMORPG**. Det är spel där man spelar en rollfigur, till exempel en riddare, som går runt i en simulerad värld och slåss med monster, samlar skatter eller bara pratar med andra spelare som spelar samma spel och går runt i samma simulerade värld. Ett par kända såna spel är **EverQuest** och **World of Warcraft**.

Redan på sjuttiotalet fanns det textbaserade liknande spel, där man (precis som i de moderna MMORPG-spelen) spelar en rollfigur, går runt i en simulerad värld, och interagerar med andra spelare. En klass av såna textbaserade spel kallas **Multi-user dungeon**, eller **MUD**. Man styrde sin rollfigur med kommandon som "gå norrut", "ta upp väskan" och "anfall björnen".

I den här uppgiften ska vi skapa en databas som kan användas för lagring av data i ett MUD.

De saker som ska lagras i databasen är följande:

1 Spelare. En "spelare" är egentligen inte en representation av själva spelaren, utan av den rollfigur som spelaren spelar i spelet. Spelaren (dvs rollfiguren) har ett unikt **namn**, en (inte nödvändigtvis unik) **beskrivning**, och några "stats", dvs värden som beskriver rollfigurens egenskaper: **styrka**, **skicklighet**, **friskhet** och hur många **poäng** som spelaren samlat ihop. Spelarna måste finnas kvar i databasen även när de loggar ut. En spelare som dör i spelet försvinner inte, utan förlorar bara sina saker och en del av sina poäng.

2 Monster. Ett "monster" behöver inte vara grönt och ha många tänder, utan alla figurer i spelet som styrs av datorn och inte av en spelare kallas för monster. Ett monster har, precis som en spelare, ett namn, en beskrivning och dessutom samma "stats" som spelarna. En skillnad är att namnet inte behöver vara unikt, utan det kan till exempel finnas många grodor som allihop heter **Groda**. Ett monster som dör i spelet försvinner.

3 Rum eller platser. Det här är de platser som spelarna och monstren kan gå omkring i. Varje spelare som är inloggad i spelet befinner sig i ett visst rum. Spelare som inte är inloggade, befinner sig inte i något rum. En del rum kan vara tomma. Varje rum har ett unikt **nummer**, ett (inte nödvändigtvis unikt) **namn** och en (inte nödvändigtvis unik) **beskrivning**.

4 Saker. Det här är de saker som finns i spelet, förutom rum, spelare och monster. Det kan till exempel vara stenar, guldkrantz, svärd och tulpaner. Sakerna kan ligga och skräpa i rummen, och dessutom kan både spelarna och monstren plocka upp sakerna och bära omkring på dem. Varje sak befinner sig antingen i ett visst rum, eller på en viss spelare eller monster. Varje sak har ett (inte nödvändigtvis unikt) **namn** och en (inte nödvändigtvis unik) **beskrivning**.

5 Servrar. Spelvärlden är så stor att den inte kan hanteras av en enda dator, utan den är uppdelad på ett antal olika datorer, eller servrar. Varje rum hör till en viss server. Spelarna och monstren kan ändå flytta sig mellan servrarna. Varje server har ett **IP-nummer** (dvs dess Internet-adress), som kan ändras ibland men som är unikt vid varje tillfälle.

De flesta mudden var ganska små, och kördes på en enda server. De innehöll kanske tio tusen rum eller platser, och några tiotal samtidiga spelare. Men nu tänker vi oss att vi ska konkurrera med EverQuest och World of Warcraft, så vi vill kunna hantera hundratals servrar, miljoner rum, och tiotusentals spelare.

Övning 2: The Universitetet

Antag att universitetet behöver ett databassystem för att hålla rätt på studenter som går kurser, vem som ger vilka kurser och var de personerna är anställda (vilken institution).

De saker som ska lagras i databasen är följande:

För att representera studenter behöver vi lagra namn (förnamn och efternamn skiljs så att man enkelt kan sortera på efternamn), personnummer för att få ett unikt id, kontonamn och lösenord. Kurser har kurskoder, namn, ges en viss period och ägs (ansvaras för) av en viss institution. De ger ett visst antal poäng och hålls av någon som är anställd på högskolan. Olika år kan en viss kurs ges olika läs-perioder och av olika personer (dock max en gång per år). Om en kurs byter namn eller omfattning i poäng, eller flyttas till annan institution måste den också byta kurskod, dvs formellt blir det en ny kurs.

Institutionen som ansvarar för kurserna är inte nödvändigtvis samma institution som läraren är anställd på.

The extra mile:

Antag att rektor utlyser pengar för pedagogiska projekt, som institutionerna kan arrangera. Man vill kunna söka efter alla olika projekten, kolla deras tidsplaner och budgetar. Projekten identifieras med namn, MEN man kan inte anta att namnen är helt unika. Inom varje institution finns en kontroll att namnen är unika, men institutionerna pratar inte med varandra.

Övning 3: That Företaget

Antag att ett företag med s.k. matrisorganisation (personalen är anställd på avdelningar, som har avdelningschefer, men arbetet organiseras i projekt där man plockar in folk från olika avdelningar efter behov) behöver hjälp att hantera information om sina anställda för lönehantering och arbetsplanering, samt att hantera de löneformåner i form av familjeförsäkringar som de anställda har. Avdelningschefers lön är delvis beroende av hur länge de varit chefer för sina avdelningar.

De saker som ska lagras i databasen är följande:

Företaget består av ett antal avdelningar. Varje avdelning har ett namn, ett nummer, en chef och ett antal anställda. Startdatum för varje avdelningschef registreras. En avdelning kan ha flera lokaler. Lokaler identifieras endast med sina namn.

Varje avdelning finansierar ett antal projekt. Varje projekt har ett namn, ett nummer och en lokal där man arbetar.

För varje anställd lagras följande information: namn, personnummer, adress, lön och kön.

En anställd jobbar för endast en avdelning men kan jobba med flera projekt som kan tillhöra olika avdelningar. Information om antalet timmar (per vecka) som en anställd planeras jobba med ett projekt lagras. Facket har drivit igenom ett krav på att ingen får beläggas mer än 40 timmar i veckan. Information gällande den anställdes avdelningschef behöver också finnas.

För varje anställd lagras information om familjen av försäkringsskäl. För varje familjemedlem lagras förnamn, födelsedatum, kön samt typ av relation till den anställda (make/maka resp barn). Man kan inte anta att informationen om familjemedlemmarna är unik. Man antar att anställda inte är gifta med varandra (vilket problem skulle kunna uppstå om vi inte antog detta).

Man ska kunna svara på följande frågor (med SQL) :

Lisa Ohlsson är sjuk idag, sök ut alla projekt där hon jobbar så att man kan skriva på lokalens whiteboard att hon är sjuk.

De anställda som har barn under 12 år ska få ett erbjudande om barnförsäkring (lista anställda med barn under 12)

Projekt X har drabbats av förseningar och behöver komma ikapp. Deltagarna behöver kunna lägga mer tid på projektet under den närmaste månaden. De avdelningschefer som har folk som jobbar på projekt X ska sammankallas för förhandlingar.

Övning 4: Check the Varuhuset

Antag att ett större varuhus-företag med butiker över hela landet behöver hålla rätt på personalen och varorna, samt kunder som får hemleveranser. Man behöver hålla rätt på personalens löner och för varorna gäller det lagersaldo och leveranser från olika leverantörer (en viss vara kan levereras av olika leverantörer).

De saker som ska lagras i databasen är följande:

Varuhusföretaget har anställda, med namn och lön, som arbetar på varuhusets olika avdelningar (namn och nummer), där man säljer olika varor (namn och nummer). Varje avdelning har en chef, som är en av de anställda.

Varorna levereras av olika leverantörer (namn och adress), och flera leverantör kan leverera samma varor, men till olika priser, som också kan variera från gång till gång.

Varuhuset har hemkörningsservice. Kunder som har konto hos varuhuset och anmält en adress för leveranser kan beställa varor och få dem levererade hem. Varje sådan beställning har ett ordernummer och ett orderdatum, utöver listan av ingående varor (naturligtvis kan man beställa mer än en av varje vara vid ett visst tillfälle).

Övning 5: Sjukhusets förlossningsavdelning

Man dokumenterar förlossningar, varje förlossning involverar en mor och ett barn som föds (som inte kan identifieras med något personnummer ännu, utan identifieras med hjälp av modern och vilken tid förlossningen startade). Flerbarnsfödlar dokumenteras som olika förlossningar med olika tidpunkter (förlossningen för en yngre tvilling anses starta då den första tvillingen är ute). Personalen som medverkar är alltid en ansvarig barnmorska och minst en men oftast flera andra sköterskor. Det finns också alltid en ansvarig förlossningsläkare, men vid en normal förlossning blir denna aldrig involverad. All personal identifieras med anställnings-ID, och man lagrar namnet och arbetspasset (dag/kväll/natt) som arbetats. Barnmorskor och läkare har också personsökare, vars nummer man lagrar.

Man vill för varje förlossning veta vilken personal som var involverad och vilket barnets status på APGARskalan var. Förlossningsförlloppet dokumenteras i kvinnans journal men om förlossningsläkaren ingriper skriver denne dessutom i sin egen dokumentation, där varje inlägg tidsstämplas men annars inte särbehandlas (dvs det är ett enda dokument som tillhör läkaren).

Övning 6: Värsta Mäklarfirman

Antag att en mäklarfirma behöver hjälp att hålla ordning på sina försäljningsobjekt, kunder och budgivning. Man vill också hålla ordning på de banker kunderna har kontakt med.

De saker som ska lagras i databasen är följande:

För varje objekt (fastighet) lagrar man adress, område, beskrivning, bild, boyta samt vilken typ av fastighet det är (lägenhet, villa, kedjehus). En viss mäklare (en av de anställda) är huvudansvarig för varje objekt. Varje objekt tilldelas en unik kod. Information om ägarna lagras också.

Varje mäklare har ett unikt ID, ett kontor och ett mobilnummer, och namn. Varje mäklare ansvarar för ett antal försäljningsobjekt.

Man lagrar också information om lånegivande banker/institut. För varje långivare (som har unika namn men också ges ett ID) har man en kontaktperson och mobilnummer till den personen, samt ett centralt telefonnumret till företaget. Man behöver ibland kontakta dessa långivare för att bekräfta bud.

Varje kund registreras med namn, adress och mobilnummer, samt hemtelefon och arbetstelefon.

Man registrerar också alternativa kontaktpersoner (t.ex maka/make) med telefonnummer. En kund ges ett kundnummer för att kunna identifieras unikt.

Ett bud på ett objekt har ett visst belopp och läggs vid en viss tidpunkt, som markeras av en unik tidsstämpel för att säkert kunna visa i vilken ordning buden lagts på ett visst objekt. Ett bud på ett objekt kan inte vara lägre än ett tidigare bud. Ett bud läggs av en viss kund via en mäklare och stöds av en lånegivare. Ett bud måste förmedlas via en mäklare, men det behöver inte vara den mäklare som har huvudansvaret för försäljningsobjektet. Ett bud har oftast, men behöver inte ha, en stödjande lånegivare (ifall kunden inte behöver ta lån).