

Reflektion

I början hade jag inte tagit hänsyn till att det kunde finnas flera kopior av en film, vilket var ett dåligt designbeslut. Jag hade redan gjort uppgift 1-3 när jag bestämde mig för att lägga filmkopior, vilket innebar att jag behövde göra några ändringar i tabellerna och modellen. Så här i efterhand skulle jag ha tagit med filmkopior från början, då det inte hade krävts någon större insats.

Tack vara att jag hade lagt ner mycket tid på att planera min modell, tänkt på att normalisera datan och verkligen funderat på vilken information jag ville ha var, så innebar ändringen inte jättemycket extra jobb. Jag behövde bara byta ut mina foreign keys från filmid till filmcopyid i en av tabellerna, samt uppdatera en av mina views.

Anledningen till att ändringen gick så smidigt var för att jag valt att designa filmdata så separerad från alla annan data som möjligt. Jag ville t.ex. att filmtabellen bara skulle innehålla data om filmen och inte ha någon koppling till antalet kopior i lager. Om Max-filmer tar in en ny film, så spelar det ingen roll om den kommer att finnas i 10 ex eller 1 ex, eller att lagerantalet ökar/minskar (kanske blir en skiva sönderrepad och måste plockas ur lager) – basinformationen om filmen är alltid samma.

Kopiorna valde jag istället att lägga i en egen tabell där varje film kopplas ihop med ett eller flera filmcopyid (på så sätt får varje kopia ett unikt serienummer). Det är detta nummer som jag sedan arbetat mot för att hämta ut data om uthyrningar och lagerstatus. Skulle Max-Filmer i framtiden vilja lägga till information, t.ex. om filmkopian är en blue-ray eller dvd – då behöver jag bara skapa en ny kolumn i filmkopia-tabellen som heter "discType" eller liknande, utan att peta i filmdata.

Nackdelen med min design är att den inte på nåt sätt är komplett. Det finns inga tabeller för betalningar, fakturor osv. Dessutom är inte heller filmdata fullständig – det saknas t.ex. information om språk, längd osv. Detta skulle kunna innebära att min design kanske inte är helt genomtänkt för framtida ändringar, och att det skulle krävas lite omstrukturering för att få allt att fungera.

Jag har i min modell tagit hänsyn till att en kund i samma order kan hyra fler än en film, men har inte behövt använda detta i någon uppgift. Min SP har inte heller i dagsläget någon möjlighet att lägga till fler filmer i en och samma order. Hade jag haft mer tid så hade detta varit en av de saker jag skulle vilja titta närmare på. Jag tycker dock ändå att min design har fungerat bra och gjort att det gått smidigt att lösa uppgifterna.

Val av storage engine

Jag har valt InnoDB eftersom den tillåter foreign key constraints och följer ACID (atomicity, consistency, isolation and durability) vilket ger pålitliga transactions. När det gäller möjligheten att sätta key-constraints så innebär det att InnoDB kontrollerar att en tabell som ska ha en specifik data faktiskt har datan. Den ser till så att rätt information hamnar på rätt ställe, hålls inom de specificerade ramarna, och att id:n blir unika.

Jag tänker även att det är bra att den har stöd för ACID, eftersom att det i en databas för en filmuthyrningssida kommer att ske många transaktioner: Några exempel: många kunder kan ex. hyra film samtidigt och då krävs det att lagerstatus stämmer, dvs. att transaktionen låser uthyrningen så att ingen annan kan hyra ut exakt samma kopia (isolation). Dessutom är det viktigt att databasen klarar att registrera alla uthyrningar även om något skulle gå fel (durability), t.ex. vid ev. överbelastningar när en ny film släpps,