

Fördelar och nackdelar med designen:

När jag först satte mig ner med uppgiften så skapade jag ett Excel ark med alla tabeller jag tänkte mig och vilka attribut dessa skulle innehålla. När jag sedan skapade modellen så skapade jag "en till många" relationer mellan movies - genre, movies - actors och movies - directors. När jag sedan skulle börja på mina vyer så insåg jag ganska snart nackdelen med denna design då jag ville att flera skådespelare skulle kunna vara med i flera olika filmer och samma gällde directors och genre. Detta resulterad i en uppdaterad modell där jag istället för relationen "en till många" skapade "många till många" relationer mellan movies – genre, movies – actors och movies – directors vilket i nuläget är en av fördelarna med min design då modellen är mer överskådlig och med tabellerna är det nu mer lättillgängligt att få ut data samt att lägga till data.

För att normalisera så mycket som möjligt i modellen så gjorde jag även en tabell in_stock som är en brygga mellan moviestabellen och rentaltabellen för att kunna ha flera exemplar av en film.

Nackdelen är att jag inte lyckats använda denna tabell. Jag ville få in antal exemplar av varje film i moviecollection vyn men har inte lyckats vilket egentligen gör tabellen in_stock lite överflödig.

I helhet har designen varit lättanvänd när jag jobbat med vyer och jag har inte haft några problem med att joina ihop olika tabeller.

När det gäller Storage Engine så valde jag ingen utan utgick från standard (InnoDB) då det har fungerat bra tidigare.