

Slutprojekt

Förutsättningar

- Ni **får ta hjälp med detaljfrågor**, typ “hur skriver jag för att skapa en tabell? vad finns det för funktion som formaterar ett datum?” men **inte med generella frågor** typ “hur ska jag tänka för att lösa uppgift 7? ser det ser rätt ut än så länge?” eller liknande. Detta gäller alla källor, vare sig det är klasskamrater, familjemedlemmar eller Stack Overflow. Om ni lyckas googla reda på en lämplig lösning är det okej att inspireras av den (inte kopiera rakt av) men ni får inte ställa en fråga i forum eller liknande om hur man ska lösa ett problem som någon annan svarar på åt er.
- Era resultat ska redovisas i ert forkade repo senast fredag 6 april 23:59. Jag kommer att kolla committernas time stamp.
- Ni ska använda er av git flow. För varje fråga ni besvarar ska ni skapa en feature-branch med ett lämpligt namn (typ `1_company_owns_movies` eller liknande) och committa de ändringar som krävs för att lösa uppgiften, inklusive ev ändringar i installations-filen. Skriv vettiga commit-meddelanden.
När ni är färdiga avslutar ni feature-branchen och mergar tillbaka den till `develop`. Till slut skapar ni en release-branch och när ni är färdiga avslutar ni releasen. Jag kommer att checka ut er `master`-branch när jag rättar.
- Under nästa veckas schemalagda tid kommer vi att ha föreläsningar kring de ämnen vi inte har tittat på än, men jag vill också kunna gå runt hos er alla för att se hur era projekt växer fram, vilka frågor ni brottas med och få en bättre förståelse för hur er kunskap ser ut. Detta kommer att vara en del av underlaget för er betygsbedömning. Det betyder inte att era projekt måste vara klara när jag är hos er eller att ni inte får ha problem som ni inte har löst, snarare att jag vill se hur ni i så fall tacklar problemen. Det innebär dock att om någon inte finns tillgänglig på vare sig måndag eller torsdag kommer det att saknas viktigt betygsunderlag.

Uppgift

Max har köpt en konkursdrabbad DVD-uthyrningsbutik. Hans affärsidé är att bygga en Netflix-konkurrent som hyr ut fysiska skivor. Besökaren går in på max-filmer.se och kan där chatta med en anställd, som kan se besökarens filmhistorik och ge personliga rekommendationer. Max skickar sedan ut skivan och besökaren skickar tillbaka den senast fyra dagar senare. Du ska göra en databas åt Max firma. Databasen ska kunna hålla reda på

- Filmer
 - Data om filmen, inklusive skådespelare, regissör, utgivningsår, genre
- Kunder
- Anställda

I övrigt får du skapa de tabeller du behöver för att kunna lösa uppgiften. Du ska skapa vyer som låter videofirman se följande:

1. Vilka filmer som firman äger, inklusive data om filmen.
2. Vilka filmer som finns i en viss genre.
3. Vilka filmer som är uthyrda, vem som hyrde dem (kund) och vem som hyrde ut dem (anställd).
4. Vilka filmer som har gått över tiden, dvs filmer som inte har blivit återlämnade trots att de borde vara det, tillsammans med namnet på kunden som har hyrt den.
5. En lista över alla anställda och hur många filmer varje anställd har hyrt ut.
6. En lista med statistik över de mest uthyrda filmerna den senaste månaden. Se fråga 10.
7. En Stored Procedure som ska köras när en film lämnas ut. Ska alltså sätta filmen till uthyrd, vem som hyrt den osv.
8. Gör en funktion som tar en film som parameter och returnerar olika värden beroende på om filmen är sent inlämnad eller inte. Dvs, om du matar in film nr 345 ska du få tillbaka **TRUE** om filmen är uthyrd men borde vara tillbakalämnad, annars **FALSE**. (1 och 0 funkar också om det är lättare.)
9. En Stored Procedure som ska köras när en film lämnas tillbaka. Den ska använda sig av ovanstående funktion för att göra någon form av markering/utskrift om filmen är återlämnad för sent.
10. Du ska underhålla en statistiktabell med hjälp av triggers. När du lämnar ut en fil ska det göras en notering om det i din statistiktabell. Du får inte lägga till informationen från din SP ovan, det ska skötas med triggers.

Betygskriterier

Nivå G

- Lös minst 6 av frågorna varav minst en inkluderar en Stored Procedure.
- Ha följt git flow enligt instruktionerna ovan.
- Tabellerna ska vara normaliserade.
- Ha lämnat in i tid. (Time stamp i repot gäller.)

Nivå VG

- Allt för G.
- Lös minst 9 av frågorna.
- Skriv om fördelar och nackdelar med din design. Du ska kunna motivera dina val. Worddokument, text-fil eller liknande.
- Berätta vilken Storage Engine du har valt och varför.
- Ta hänsyn till att man kan ha flera av varje film, t ex 10 ex av Saving Private Ryan.

Redovisning

Redovisa med en modellfil och ett installationsskript som skapar databasen, tabellerna, vyer, stored procedures osv och all data. Databasen ska innehålla tillräckligt med data för att kunna testa vyerna ovan, minst 10 personer, 10 filmer osv.

Döp databasen till ditt förnamn och en bokstav från ditt efternamn, t ex MikaelO. Det är viktigt! Det är jättekigt att försöka hålla koll på 15 databaser som alla heter Movies. Om din databas inte följer denna regel kommer jag inte att rätta ditt projekt.

I mitt repo finns en fil som heter repo.txt. Där ska ni klistra in url:en till ert repo i https-format, inte ssh. Glöm inte att committa och pusha till master-branchen. (I mitt repo behöver ni alltså inte följa git-flow, bara i era egna.)

Innehållet i ditt repo ska se ut såhär:

- | | |
|----------------|---|
| - database.sql | Filen som skapar databasen, tabellerna etc. När jag har klonat ditt repo kommer jag att importera filen som heter database.sql. Finns inte filen kommer jag att få ett fel och kommer inte att ha något resultat från dig att titta på. |
| - queries.sql | Den här filen ska innehålla sql:en jag ska köra för att se resultatet från dina lösningar. Innehållet ska följa modellen i det klonade repot. |
| - model.mwb | Innehåller din modell. |

Du ska även inkludera en Word-fil eller liknande om du siktar på VG i enlighet med

betygskriterierna ovan.

Anledningen till att det är noga med att döpa filer och repon rätt är att jag kommer att försöka automatisera så mycket som möjligt av kloning av repon, installation mm för att minska min belastning och kunna rätta uppgifterna så snabbt som möjligt.