

inlämningsuppgift 1

Databaser och Linux

Punkter som evalueras:

Kunskaper:

1. Redogöra för databas-systemen MySQL, MongoDB samt NOSQL
2. Förklara elementen inom Databashantering
3. Ge exempel på skillnader mellan SQL och NOSQL

Färdigheter:

4. Använda SQL miljön för att bygga och hantera en SQL-databas
5. Använda NOSQL miljön för att bygga och hantera en NOSQL-databas
6. Tillämpa sökningar i olika databassystem
7. Använder sig av CRUD i sin miljö

Kompetenser:

8. Skapa, utveckla och underhålla egna samt andras databaser

*punkt nummer 8 visas inte tydligt här i uppgiften,
men det är för att principen är densamma oavsett om du använder din egen databas eller
någon annans.*

*Självklart kan ni labba med att kommunicera med en kurs-kamrats databas om ni känner för
det, men det är absolut inget krav då ni kommer visa på denna kompetens även om ni bara
använder er utav en egen databas.*

Det tillkommer även frågor längre ned som evaluerar denna kompetens extra.

(Ex, "Om du ger en annan part tillgång till din databas, vad är då viktigt att tänka på?"... med mera)

Bank-konto databas

I den här inlämningsuppgiften ska vi skriva ett litet program som kan hantera bank-konton i en databas.

Med den här filen följer det med "fake"-data som skall användas i databasen.

Det följer även med lite shell-script för att importera datan.

Del 1 (MySQL / MariaDB)

Denna uppgift skall skrivas i ".sql" filer.

"funktionerna" som nämns nedan skall skrivas i separata .sql filer.

1. En funktion för att skapa nya bank-konton i databasen. (create.sql)
2. En funktion för att hämta alla bank-konton i databasen. (read.sql)
3. En funktion för att uppdatera "holding" på ett bank-konto med ett specifikt ID i databasen. (update.sql)
4. En funktion för att ta bort ett bank-konto i databasen. (delete.sql)

Med dessa punkter implementerade har du nu implementerat något som kallas för CRUD.

Testa att köra varje funktion för sig och se till att dom fungerar.

Del 2 (MongoDB)


Här ska vi skriva funktioner som utför operationer mot MongoDB databasen.

Dessa funktioner skall skrivas: (samma som i del 1 fast mot MongoDB).

Dessa funktioner skall skrivas i separata .sh filer.

Exempel:

update.sh



```
mongo --eval 'db.bank_accounts.update({"name":"foo"},{$set:{"this":"that"}});' bank
# "bank" i slutet är namnet på din databas.
```

1. En funktion för att skapa nya bank-konton i databasen. (create.sh)
2. En funktion för att hämta alla bank-konton i databasen. (read.sh)
3. En funktion för att uppdatera ett bank-konto i databasen. (update.sh)
4. En funktion för att ta bort ett bank-konto i databasen. (delete.sh)

Med dessa punkter implementerade har du nu implementerat något som kallas för CRUD.

Testa att köra varje funktion för sig och se till att dom fungerar.

Del 3

Skapa upp en ny tabell / collection i dina databaser, både i MongoDB och i MySQL / MariaDB.

Denna tabell skall heta "locations".

Låt denna tabell sedan bara ligga där, vi kommer att använda oss utav den i inlämningsuppgift 2.

Anledningen till att vi skapar upp tabellen i denna inlämningsuppgift är för att visa på delar av kunskaperna i punkt 4 och 5.

Frågor

1. Var hittar man log-filerna för databaserna? (MySQL / MariaDB, och MongoDB)
2. Vad är skillnaden mellan en SQL databas som MySQL / MariaDB och en NoSQL databas som MongoDB?
3. Vad finns det för likheter mellan databaserna?

4. Fundera, när vill man välja en SQL databas, och när vill man välja en NoSQL databas?
5. Om du ger en annan part tillgång till din databas, vad är då viktigt att tänka på?
6. Vad för typ av information kan vara känsligt / problematiskt att spara in en databas när det kommer till tex GDPR?
7. Nämn några SQL-databaser, samt några NoSQL-databaser.
8. Om flera företag, organisationer eller personer använder samma databas, varför är det då viktigt att sätta upp separata användare i databasen för dessa parter?
9. Vilka delar i CRUD påverkar / gör ändringar i databasen, och vilka delar i CRUD gör inte det?
10. Hur ansluter man till någon annans databas från terminalen? (Visa både på MySQL och MongoDB)