

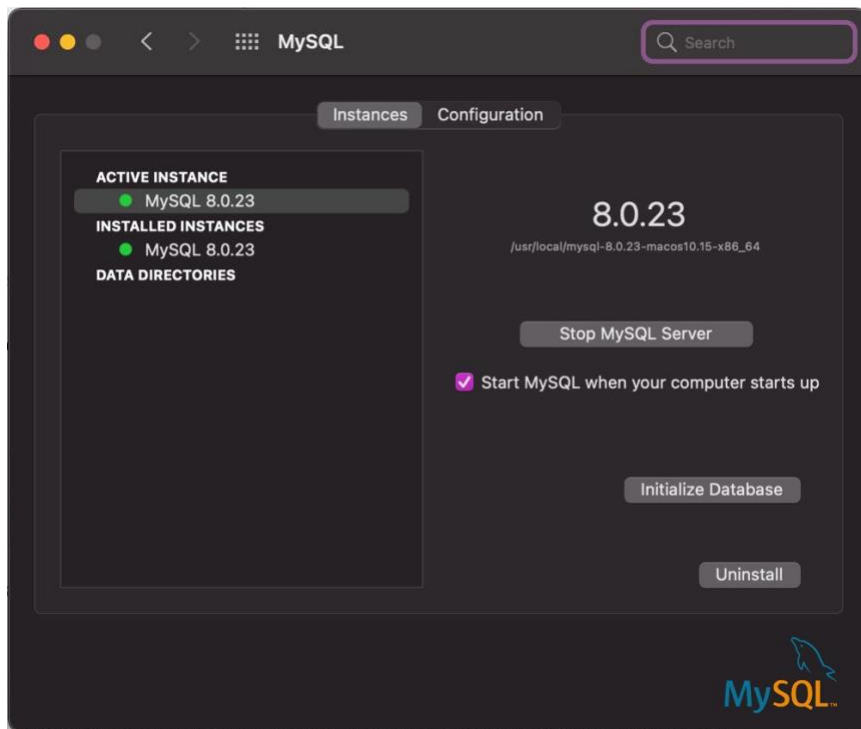
## MAC instruktioner för att komma igång med MySQL i Netbeans

Program och filer du behöver ha nerladdat på datorn

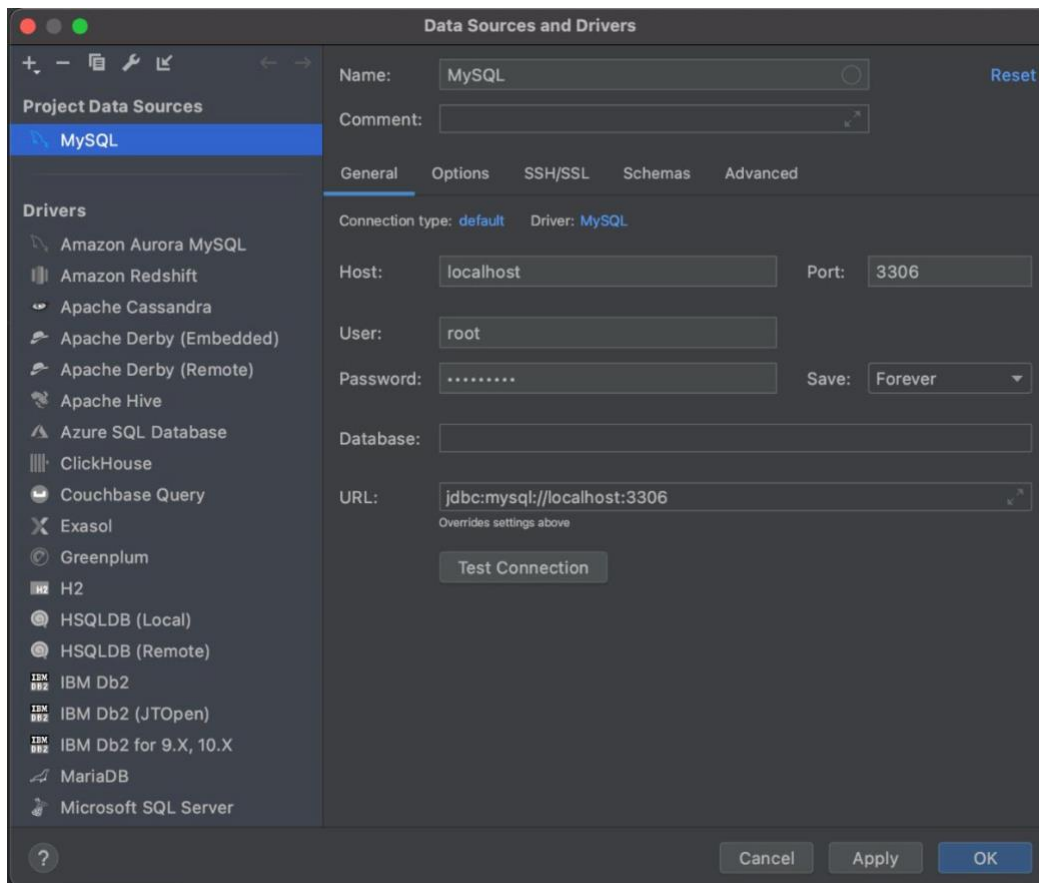
- **SQL Community Server:** <https://dev.mysql.com/downloads/mysql/>
- **Datagrip:** <https://www.jetbrains.com/datagrip/download/#section=mac>
- **Netbeans IDE:** <https://netbeans.apache.org/download/nb122/nb122.html>
- **Databasfiler.zip** från Blackboard

### STEG 1: Skapa databasen "mibddb" i Datagrip

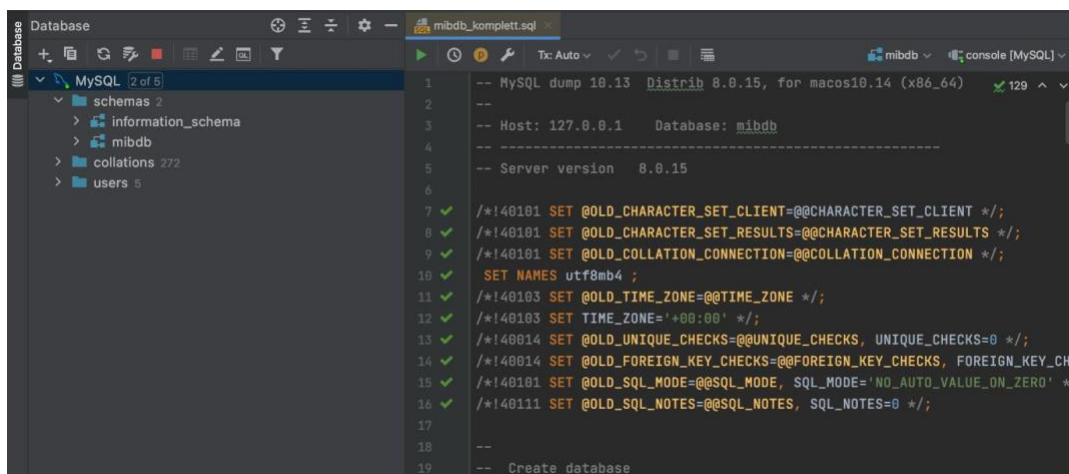
Se till att MySQL servern är igång på din dator. Du kan kolla inställningarna i applikationen **systeminställningar** som finns på MAC (versionsnumret kan skilja)



1. Skapa en ny **Data Source** i Datagrip och välj **MySQL**. Ge den valfritt namn, det kommer inte användas mer än att det är det namn det listas med i Datagrip efteråt. **Host** ska vara "localhost" och **port** ska vara "3306" (porten kan vara något annat också, men ändra inte om inte du måste). **User** är namnet som du antingen tilldelades automatiskt eller fick välja när du installerade MySQL. Valde du inte namn på din MySQL user så är det "root" som gäller. **Password** är det du valde när du installerade MySQL. Lämna **Database** tomt. Klicka på **Test Connection** och lyckas det kan du klicka på **OK**.



- Öppna ett nytt query-fönster i Datagrip, lägg in filen **mibdb\_komplett.sql** som finns i mappen **Databasfiler** du laddat ner från blackboard och kör frågan via din **Data Source**. Efteråt borde databasen **mibdb** dyka upp under mappen **schemas** i din nyskapade Data Source (se bild nedanför). Efter detta kan du stänga ner Datagrip.



Ps. Om du tittar på skriptet för **mibdb\_komplett.sql** ser du att det ganska högt upp skapas en **user** för mibdb-databasen. Dessa uppgifter är det som vi kommer använda sen för att koppla upp till databasen via Netbeans vi MÅSTE således köra den koden **mibdb\_komplett.sql** först så att databasen och användaren för att koppla upp sig mot databasen faktiskt finns i MySQL

```
CREATE USER mibdba@localhost IDENTIFIED BY 'mibkey';
```

## STEG 2: Importera JAR-filerna som behövs för att kunna börja koda mot databasen

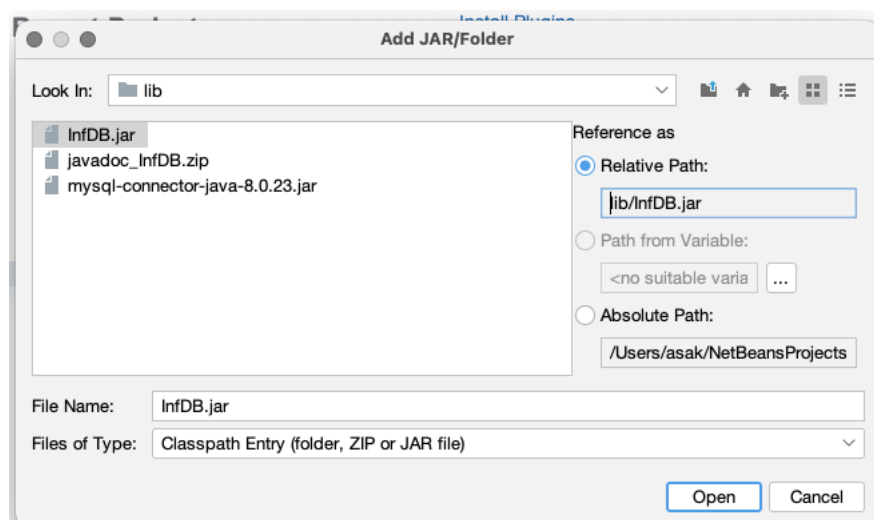
3. Ladda ned filen "Databasfiler". Den innehåller ett antal filer som ni kommer att använda för att skapa en uppkoppling till en MySQL databas.
4. När du skapat ett projekt i netBeans ska du i root-mappen för det projektet även skapa en till undermapp som heter "lib".

> build	4 April 2023 at 14:44	--	Folder
build.xml	4 April 2023 at 14:05	4 KB	XML
> dist	4 April 2023 at 14:44	--	Folder
> lib	4 April 2023 at 14:09	--	Folder
manifest.mf	4 April 2023 at 14:05	82 bytes	Document
> nbproject	4 April 2023 at 14:05	--	Folder
> src	4 April 2023 at 14:05	--	Folder

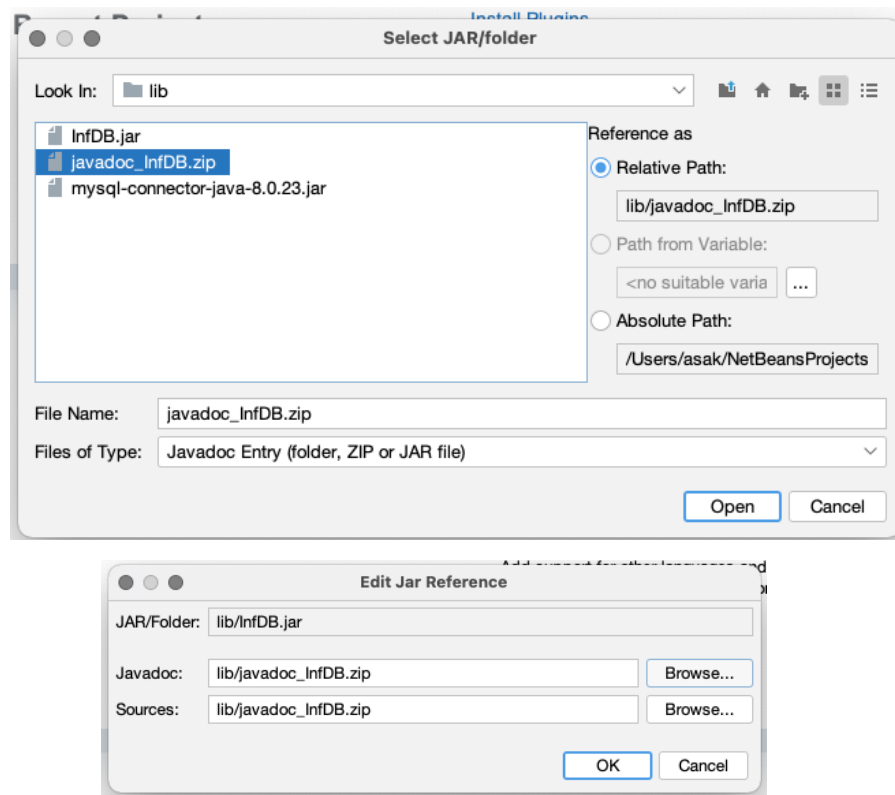
5. Packa upp/kopiera in filerna från den nedladdade "Databasfiler" till "lib" mappen i ditt projekt

InfDB.jar	23 February 2021 at 17:26	5 KB	Java JAR file
javadoc_InfDB.zip	23 February 2021 at 17:26	44 KB	ZIP archive
mibdb_komplett.sql	23 February 2021 at 17:26	16 KB	SQL
mysql-connector-java-8.0.23.jar	21 February 2021 at 23:10	2,4 MB	Java JAR file
README.txt	24 February 2021 at 19:29	589 bytes	Plain Text

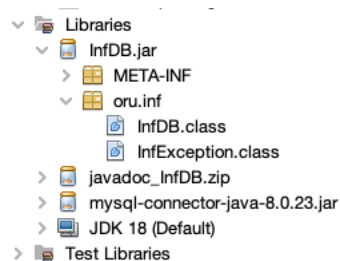
6. Detta kommer underlätta delning (ex. via Github) av projektet mellan gruppmedlemmar då alla filer som kommer behöva användas i projektet kommer kunna ha relativa sökvägar. Alla gruppmedlemmar behöver således inte ha alla filer själva på sin dator någonstans med absoluta sökvägar (ex. C:\user\anna...) som andra datorer naturligt inte kommer åt, utan de finns i stället tillgängliga i projektmappen.
7. Databasklassen finns i ett paket som vi måste lägga till i projektet vilket förs genom att gå in på "libraries" i ert projekt i netBeans. Högerklicka och väljer "Add JAR/Folder" och letar fram till projektmappen och väljer "InfDB.jar". Klicka på "Relative Path" som återfinns till höger i "Add JAR/Folder" vyn var det då ska stå "lib/InfDB.jar" likt den blåa rutan i bilden nedan.



8. Därefter gör ni på samma sätt för "mysql-connector-java-8.0.23.jar" och "javadoc\_InfDB.zip"
9. Därefter högerklickar ni på "InfDB.jar" i Libraries och tar "edit" för att där koppla "javadoc\_InfDB.zip" till databasklassen "InfDB.jar" likt nedan



10. Med "InfDB.jar" får vi två klasser. Dels databasklassen "InfDB.class" men även en undantagsklass som hanterar undantag från just vår databasklass



### STEG 3: Skapa databasuppkoppling mellan ditt projekt och MySQL-databasen

11. För att kunna skapa en koppling mot mySQL så behöver importera dem klasser som kommer att hantera uppkoppling likt nedan

```
import java.util.logging.Level;
import java.util.logging.Logger;
import oru.inf.InfDB;
import oru.inf.InfException;
```

12. Vi kommer även att behöva ha ett privat fält av typen InfDB likt nedan

```
public class InloggningsFonster extends javax.swing.JFrame {

    private InfDB idb;
```

13. En uppkoppling mot mySQL databas måste alltid kodas innanför ett try-catch block. Skälet till det är att programmet inte kan veta om vi har en databas tillgänglig eller inte. Kan ex. vara att det uppstått ett fel i dbms som innebär att ingen databas körs på datorn i bakgrunden. Kan vara att vi anger felaktiga användaruppgifter. InfDB klassen kommer att kasta undantag vid fel som således måste fångas upp och hanteras för att inte programmet ska krascha. Därav importen av av InfException, Level och Logger ovan utöver InfDB klassen. Således kodar vi följande (mer förklaring på nästa sida):

```
try {
    String dbSchema = "x";
    String dbPort = "3306";
    String dbUsername = "y";
    String dbPassword = "z";

    idb = new InfDB(dbSchema, dbPort, dbUsername, dbPassword);
} catch (InfException ex) {
    Logger.getLogger(Start.class.getName()).log(Level.SEVERE, null, ex);
}
```

Exempel:

```
package testapplikation;

import java.util.logging.Level;
import java.util.logging.Logger;
import oru.inf.InfDB;
import oru.inf.InfException;

public class TestApplikation {

    private static InfDB idb;

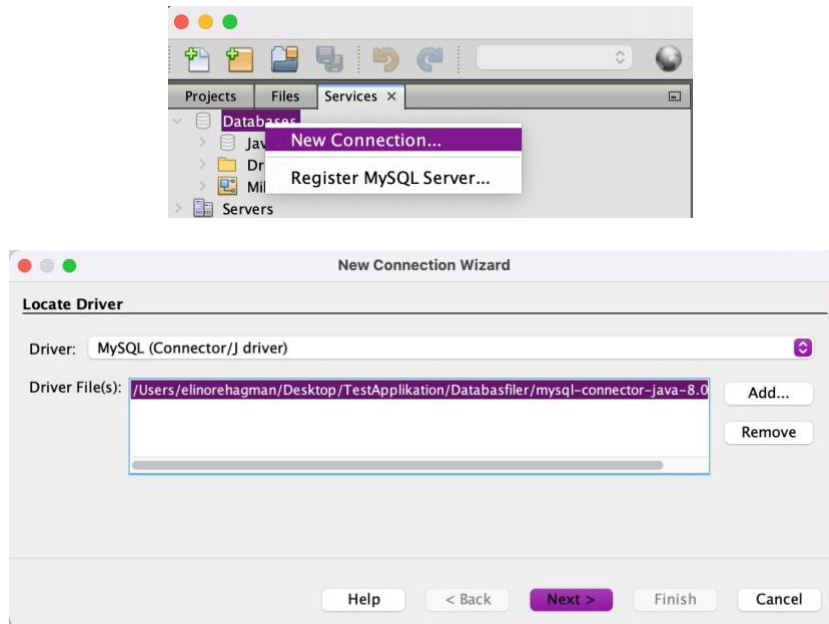
    public static void main(String[] args) throws InfException {

        try {
            idb = new InfDB("mibdb", "3306", "mibdba", "mibkey");
        } catch (InfException ex) {
            Logger.getLogger(TestApplikation.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

14. Vi behöver instansiera ett objekt av typen `InfB`. Konstruktorn för `InfDB` har fyra `String` parametrar som motsvarar följande (i ordning från första till sista paramenter):
- ***Databasschemanamnet*** där databasen vi vill koppla upp oss mot återfinns
  - ***Den port som MySQL använder för uppkopplingar*** vilket för de flesta kommer vara "3306" kan dock skilja för dig. Du valde vilken port vid installation, kan ses om ni kollar properties i er uppkoppling mot MySQL i Datagrip
  - ***Användarnamn*** för användaren som har admin. rättigheter till databasen (kommer vara ett speciellt användarnamn för uppgiften på delkurs 4)
  - ***Lösenord*** för för användaren som har admin rättigheter till databasen (kommer vara ett speciellt lösenord för uppgiften på delkurs 4)
15. Vi kan nu använda samma databaskoppling genom att skicka med `idb` som en parameter till alla klasser som behöver databas-access. Där dessa klasser kommer behöva ett fält av typen **`private InfDB idb;`** vari det i de klassernas konstruktor har åtminstone 1 parameter av typen `InfDB` där vi i konstruktorn sedan kodarföljande: **`this.idb = idb;`**

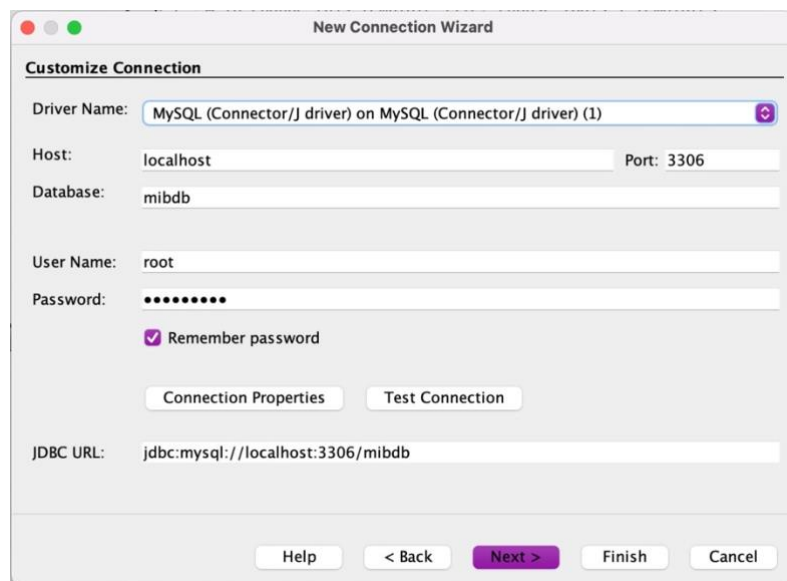
## STEG 4: Använda netBeans för att koda mySQL i stället för Datagrip eller Workbench

16. Klicka på **Services** uppe i projektmenyn, högerklicka på **Databases** välj **New Connection**. Först behöver vi lägga till en MySQL driver, så klicka på **New Driver** i dropdown-menyn. Klicka sen på **Add** och välj **mysql-connector-java-8.0.23.jar** som du hittar i mappen **lib** för ditt Java projekt och klicka på **OK**. Klicka sen på **Next**.



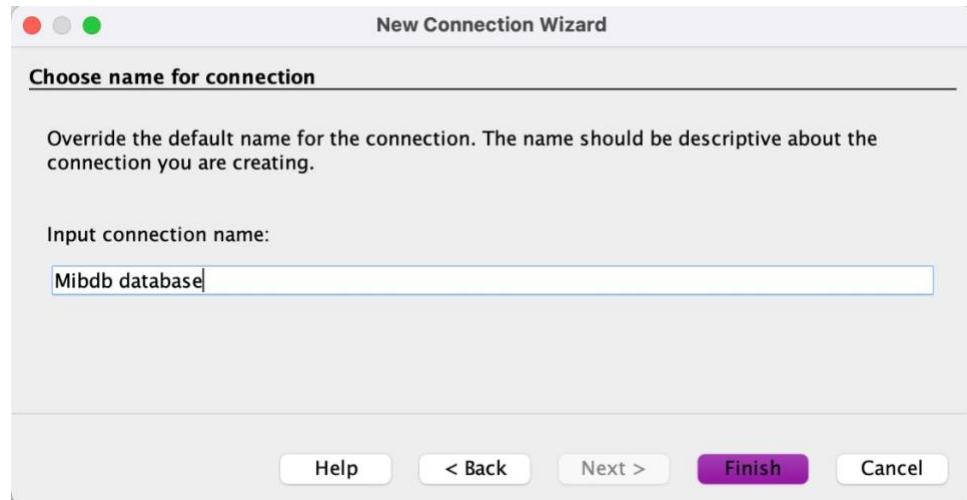
Nu är det dags att fylla i informationen som behövs för att Netbeans ska kunna koppla upp sig mot databasen vi skapade i Datagrip. **Database** är namnet på databasen som du kan se i sidomenyn på Datagrip, alltså **"mibdb"**. **User Name** och **Password** är den **user** som skriptet skapade, alltså **"mibdba"** och **"mibkey"**.

Se även till att **JDBC URL** är: **jdbc:mysql://localhost:3306/mibdb**

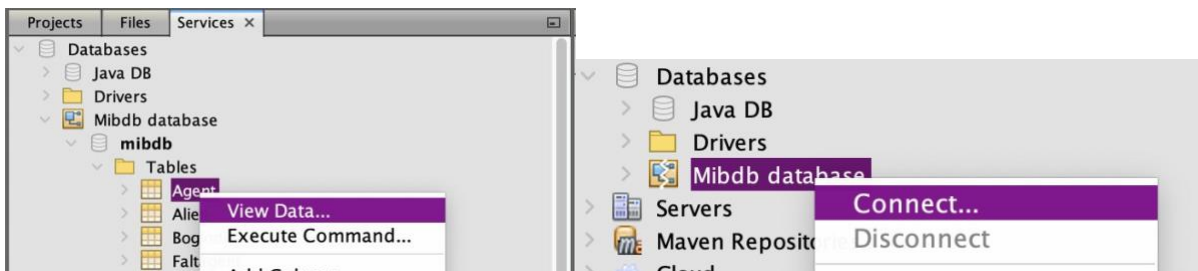


17. Klicka på **Test Connection** och se ifall uppkopplingen lyckades.

18. Om det gick bra så är det bara **Next** tills du kan skriva in **Input connection name**. Där får du möjlighet att ge ett namn till databasen (endast för att särskilja olika databaser i Netbeans sidomeny ifall du har flera, ingen mer praktisk funktionalitet), och klicka sen på **Finish**.



19. Efter det kommer databasen dyka upp under **Databases**, och du borde kunna gå in och titta på tabellerna och köra SQL-frågor mot den. Har databasen en trasig ikon och inte har något innehåll när man försöker öppna upp den (kanske händer under projektets gång) är det bara att högerklicka och försöka återkoppla.



20. Efter att vi "connectat" till db kan vi således välja "execute Command" och här testa SQL-kode för att sedan commita dem och se var vi får för data ut. När vi sedan är nöjda med att SQL-koden fungerar som tänkt kan vi därefter använda dem i vår Javakod.