

# **Mysql**

# **Databashanterare**

# 1. Abstract/sammanfattning

Denna rapport är dokumentations delen för mitt slutprojekt i programmering 2 och programmet som skrevs är en Mysql databashanterare för att möjliggöra kommunikation med databaser för att utföra grundläggande CRUD uppgifter men även mer avancerad SQL administration och manipulation.

# 2. Innehållsförteckning

<b>Abstract/sammanfattning</b>	<b>2</b>
<b>Innehållsförteckning</b>	<b>2</b>
<b>Syfte/frågeställning</b>	<b>2</b>
<b>Teori, bakgrund</b>	<b>3</b>
SQL	3
SWING	3
<b>Beskrivning av lösningen</b>	<b>3</b>
Användarhandledning	3
För slutanvändaren	3
För Utvecklare	5
JavaDoc/API	6
Systembeskrivning/klassdiagram	6
<b>Testkörning</b>	<b>7</b>
<b>Diskussion</b>	<b>8</b>
<b>Bilagor</b>	<b>9</b>

### 3. Syfte/frågeställning

Syftet med denna uppgift är att göra ett eget projekt och därför så har jag valt att göra en sql database hanterare. Detta program är ett gui som används för att kommunicera med olika databastyper. Programmet är skrivet med objektorienterad programmering i högsta spets så även om det endast stödjer mysql i dagsläget så är det inte mycket som behöver läggas till för att den ska stödja fler databastyper. Detta gör att vidare expansion eller ändring av programmet är väldigt lätt att genomföra.

### 4. Teori, bakgrund

Anledningen till att verktyget behövs är för att jag inte kunde välja mellan de olika alternativen för att kommunicera med databaser så som Sequel Pro, Datagrip, phpmyadmin och mysql workbench. Detta resulterade i att jag bestämde mig för att skriva ett eget program.

- **SQL**

Sql är ett query språk som primärt används till databashantering. Det utvecklades av några forskare från IBM under 70-talet och är en internationell standard som används världen runt inom programmering och datorteknik.

- **SWING**

Swing är ett bibliotek som kommer inbyggt i java som är för att bygga grafiska användargränssnitt som är en efterträdare av AWT som också var ett bibliotek inbyggt i java.

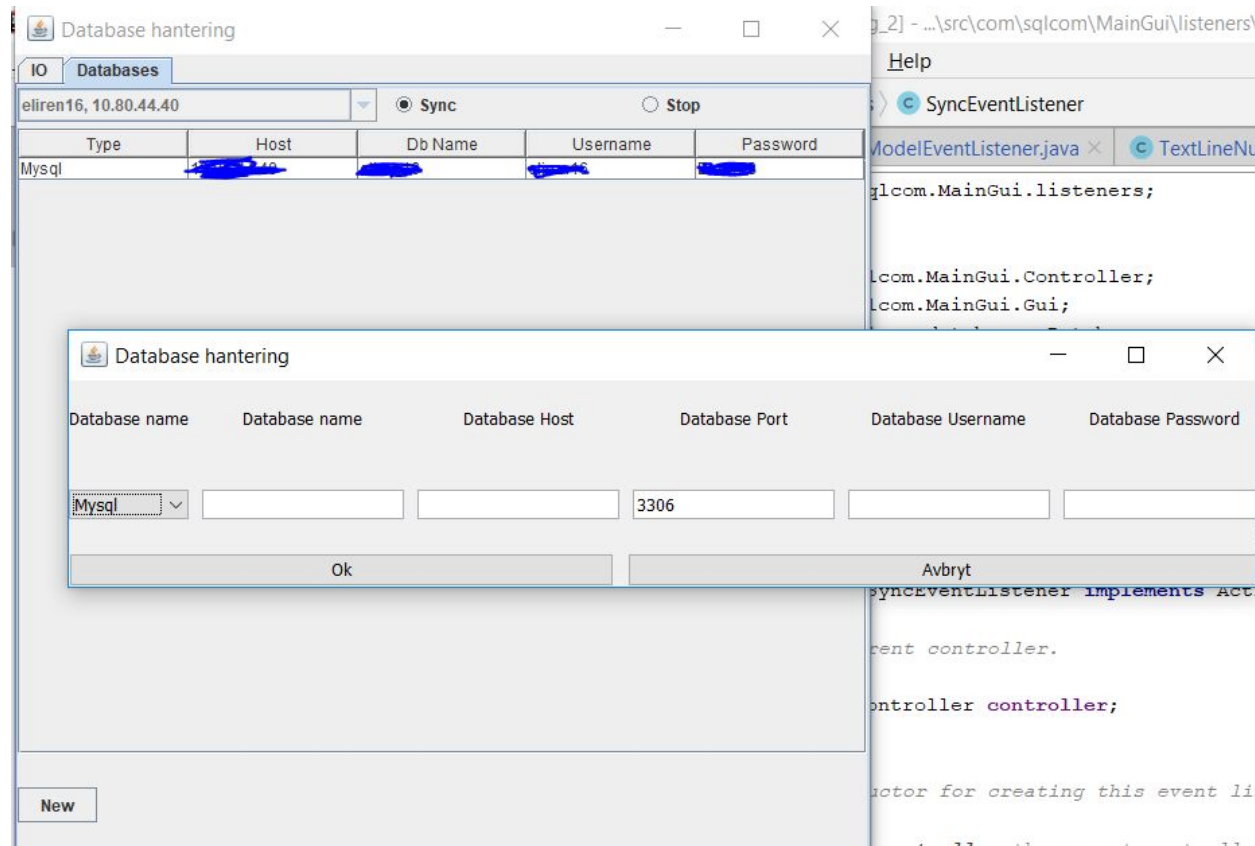
### 5. Beskrivning av lösningen

- **Användarhandledning**

Då projektet har öppen källkod så finns det två huvudsakliga alternativ för att köra programmet, som slutanvändare och som utvecklare, nedan förklaras hur man laddar ner och startar båda.

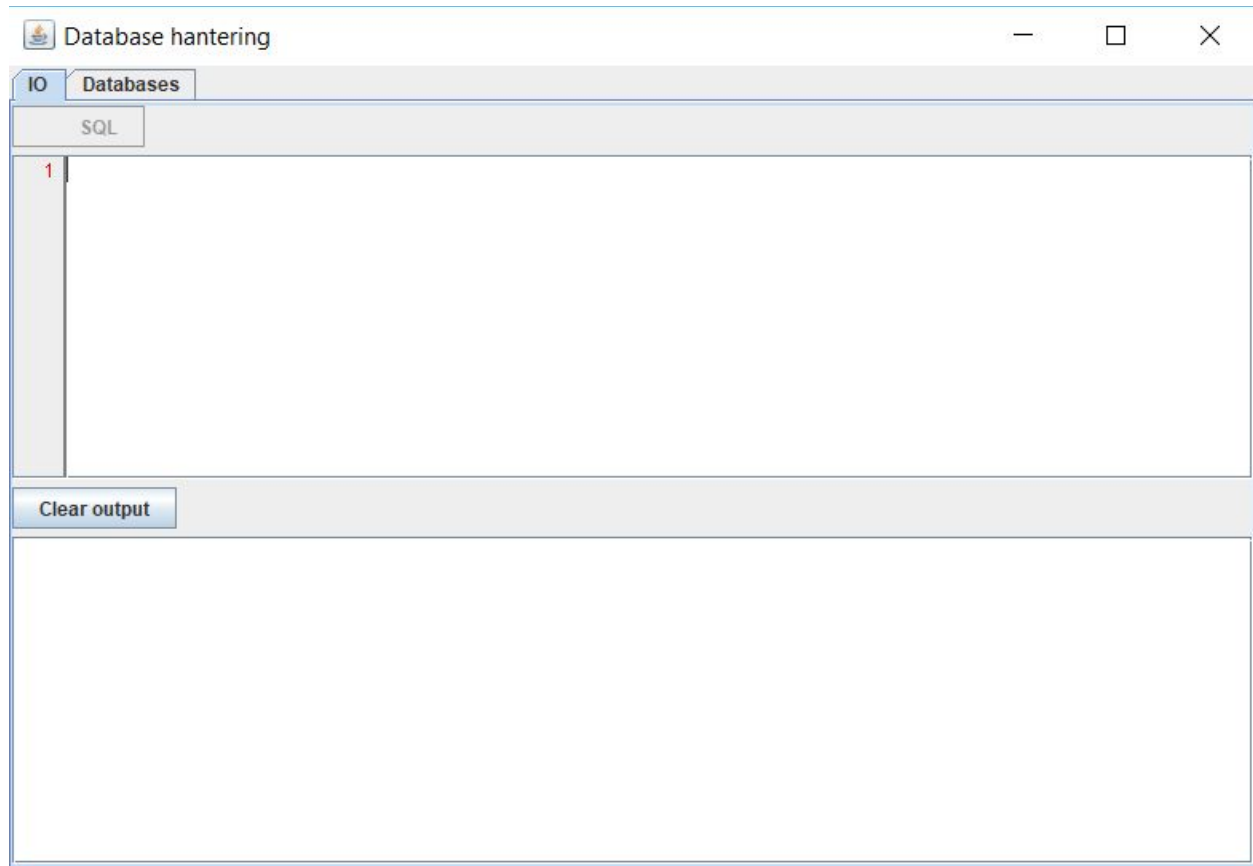
## För slutanvändaren

För att starta programmet så behövs endast jar filen som finns [här](#). Java JRE behövs också i minst version 1.8 eller JAVA 8. då filen är nedladdad så är det bara att dubbelklicka på den och köra, för att lägga till databaser att koppla upp sig mot så går du under tabben “Databases” och klickar på new, då borde en ruta visa sig. Denna ruta fyller du in datan som efterfrågas och ser ut såhär.



Klicka “OK” den kommer att läggas till i en tabell och i dropdown menyn. Efter du lagt till en databas och vill ändra datan så klickar du på rutan du vill ändra i tabellen. skriver din ändring och trycker enter.

För att öppna en uppkoppling till databasen så klicka du på dropdown menyn uppe i vänstra hörnet, väljer din databas sedan klickar på knappen “Sync”. Efter detta så går det att gå till IO tabben och börja skriva sql i den övre rutan.



I den nedre rutan kommer din output ut eller annan information. För att rensa ut rutan så klicka på knappen “Clear output”. Kom ihåg att om “Run sql” knappen är avstängd så har du inte kopplat upp dig till databasen och måste välja databas på andra tabben och klicka på “Sync” knappen.

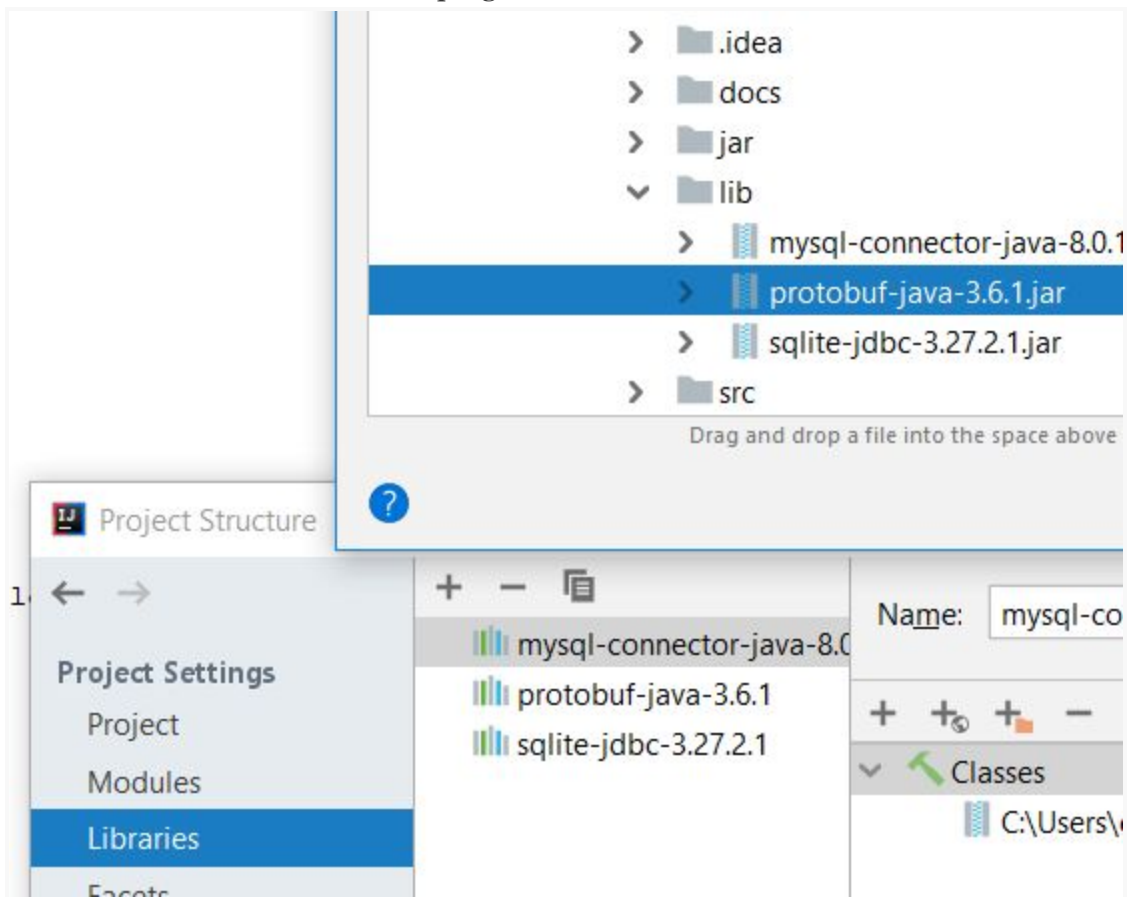
## För Utvecklare

För att jobba vidare på programmet eller klona det så börja med att skriva

```
$ git clone https://github.com/Eliasr123/Slutprojekt_programmering_2.git
```

Efter det så öppnar du projektet och lägger till alla libraries i mappen “lib”, för att göra detta i IntelliJ så öppnar du Project Structure `ctrl+shift+alt+s` eller under `File>Project Structure`, Klicka på Libraries och den lilla plus ikonen. Hitta jar filerna under lib och lägg till dem en efter en. Efter det så ska programmet starta och innehålla alla externa

bibliotek som behövs för att köra programmet.



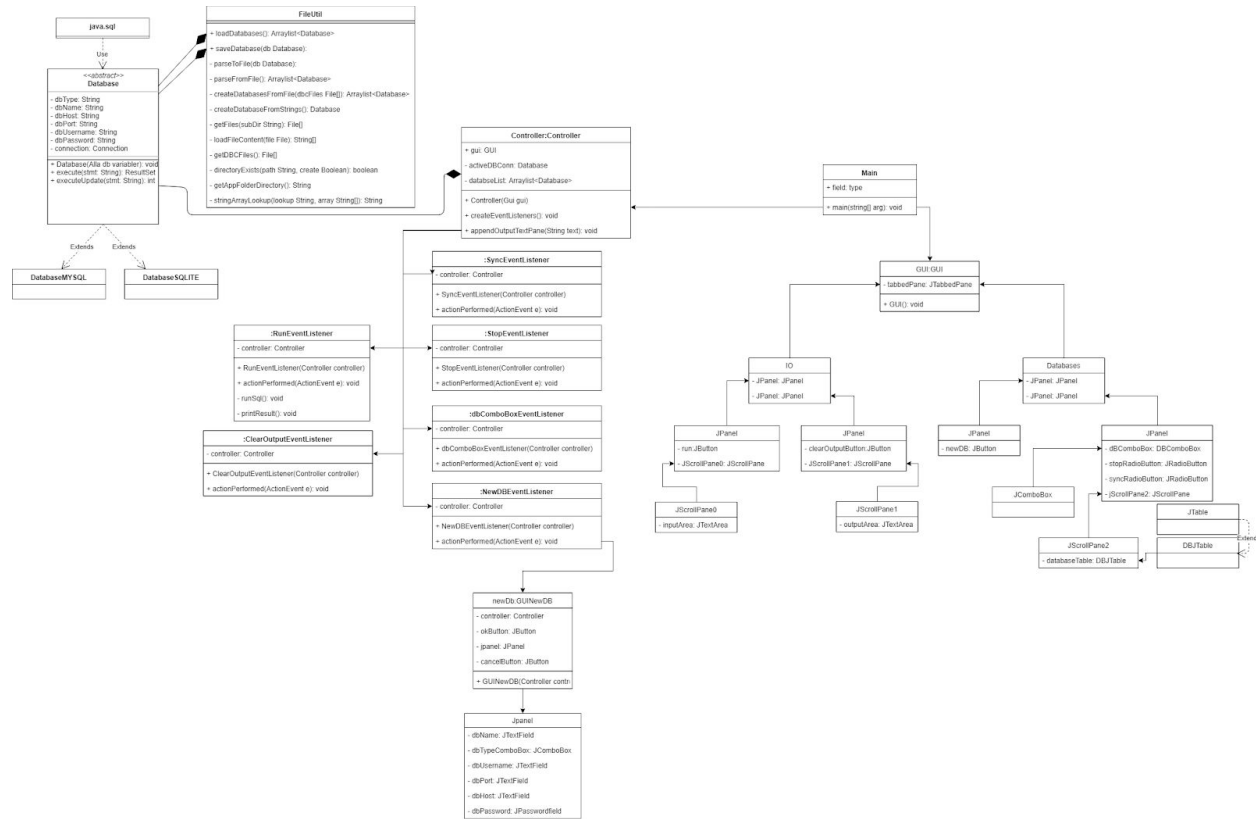
- **JavaDoc/API**

Länken för javadocs är här:

[https://eliasr123.github.io/Slutprojekt\\_programmering\\_2/](https://eliasr123.github.io/Slutprojekt_programmering_2/)

- **Systembeskrivning/klassdiagram**

Innan programmet skrevs så skapades detta uml diagram som översikt av programmet med huvudsakligen GUI komponenter på den högre delen av bilden medans logiken för backenden så att säga är på vänstra delen av bilden.



## 6. Testkörning

Programmet startar och fungerar med alla dess inplanerade funktioner. här är en skärmdump på programmet då det kört två sql queries efter varandra.

Database hantering

IO

Databases

Run SQL

1

SHOW tables;

2

SELECT \* FROM migrations;

Clear output

Tables\_in\_

armor

migrations

password\_resets

players

potion

stats

weapon

id	migration	batch
1	2014_10_12_000000_create_players_table	1
2	2014_10_12_100000_create_password_resets_table	1
3	2019_04_04_081703_create_weapons_table	1
4	2019_04_04_082123_create_armor_table	1
5	2019_04_11_090100_create_stats_table	1
6	2019_05_03_091242_create_potions_table	1

## 7. Diskussion

På grund av det extensiva förarbetet innan programmering fasen så gick själva programmerings delen väldigt bra utan några större problem. Att göra ett sånt detaljerat klassdiagram var ganska mycket arbete men resultatet var lätt värt det.

Om jag skulle göra om programmet eller vidareutveckla det så skulle jag börja med att lägga till en funktion för att möjliggöra att ta bort databas data. Utöver detta så skulle jag se om det går att implementera en autocompletion för sql queries och en mer användarvänlig GUI.



Det enda som gick dåligt var att programmet inte går att starta som en jar genom terminalen då java JRE 11 typ inte existerar och programmet är satt till java 11, att ändra target till Java 8 går inte då programmet använder sig av nya funktioner såsom lambda och en del andra nya funktioner. Detta har åtgärdats i senaste versionen av programmet så att den går att starta i Java 8.

## 8. Bilagor

[Github repot](#)

[Javadocs](#)