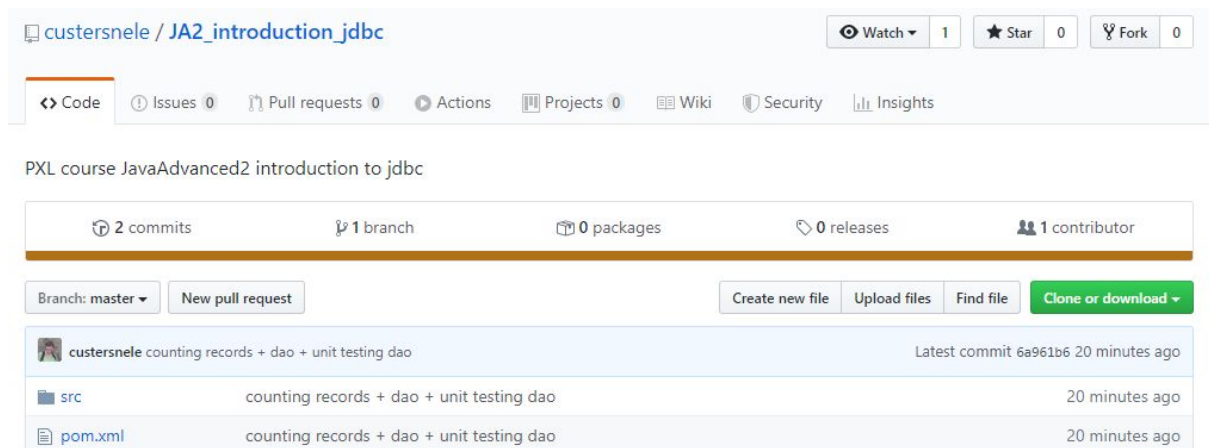


JDBC

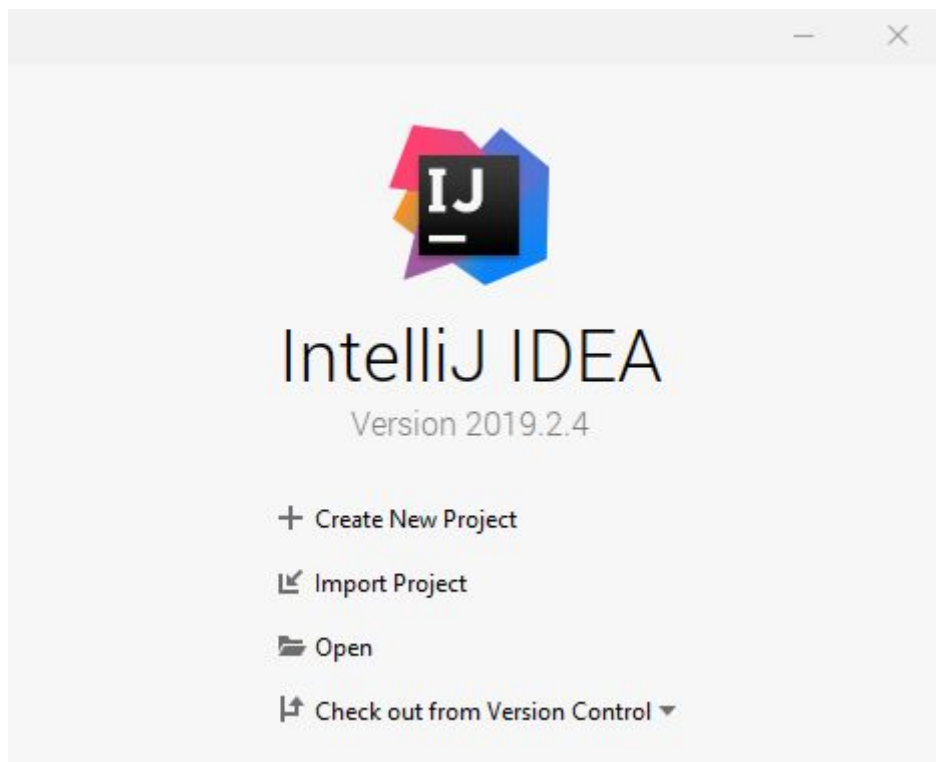
JDBC staat voor Java Database Connectivity en is de technologie die communicatie met relationele database mogelijk maakt.

0. Demo's

Ga naar de github repository https://github.com/custersnele/JA2_introduction_jdbc en fork je eigen kopie van dit project.



Open nu jouw project in IntelliJ IDEA.



Klik hiervoor op Check out from Version Control en selecteer git.

Vul de URL van jouw repository in en test eventueel de verbinding. Kies ook een directory om jouw projectcode lokaal op te slaan.



Het project wordt vervolgens geopend in de ontwikkelomgeving.

1. De databank

Er zijn 2 mogelijkheden om lokaal een databank beschikbaar te maken. De gemakkelijkste is door gebruik te maken van docker. Docker wordt gebruikt om toepassingen eenvoudig uit te voeren in 'containers'.

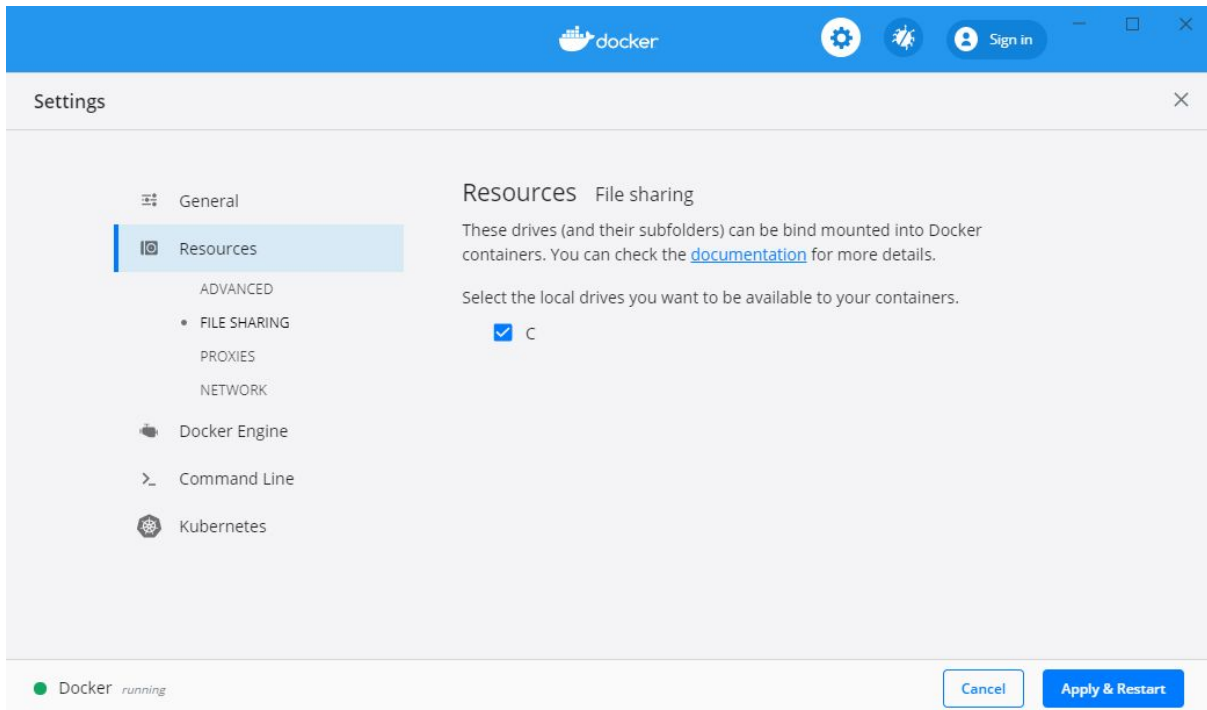
Niet iedereen met een windows systeem kan docker installeren, beschik je over Windows Home dan zal de installatie niet lukken en installeer je best lokaal een MariaDB. Ga in dat laatste geval direct naar 1.2 Installatie MariaDB.

1.1 Installatie docker

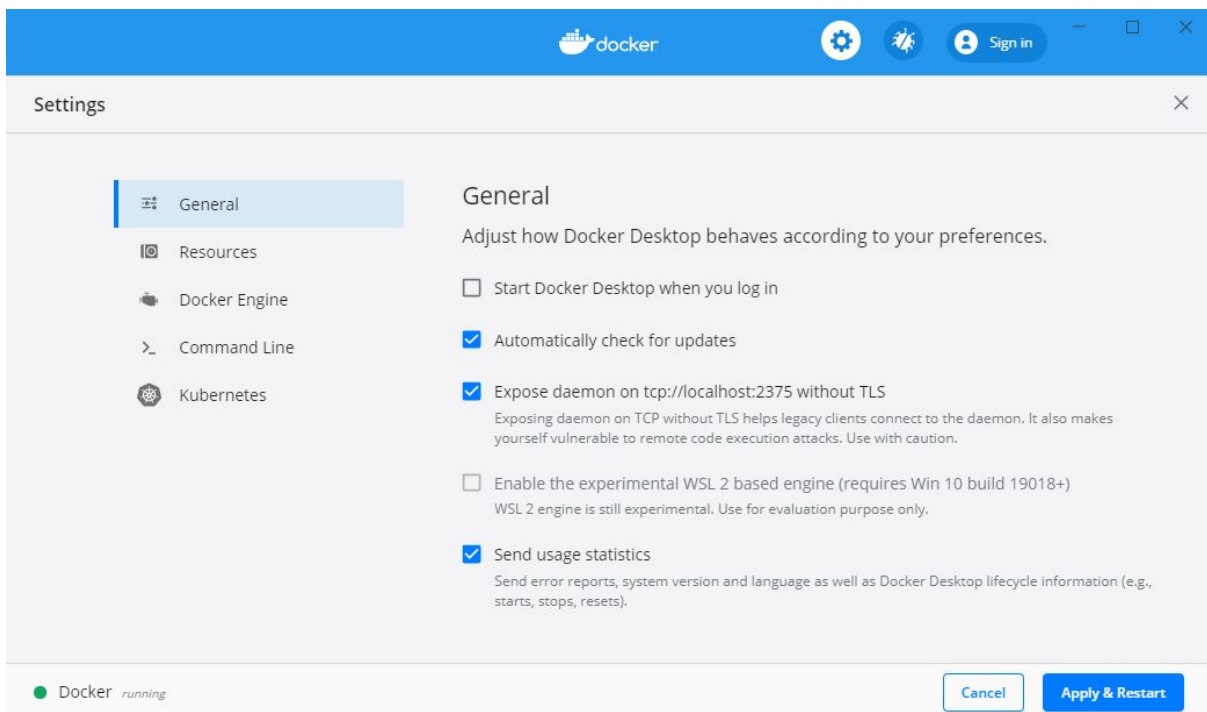
Download en installeer docker: <https://docs.docker.com/docker-for-windows/install/>

Na de installatie passen we enkele instellingen aan (windows):

1. Resources > File sharing
Maak je C drive beschikbaar voor je containers.



2. Laat IntelliJ toe om verbinding te maken met docker
General > Expose daemon on tcp://localhost:2375 without TLS

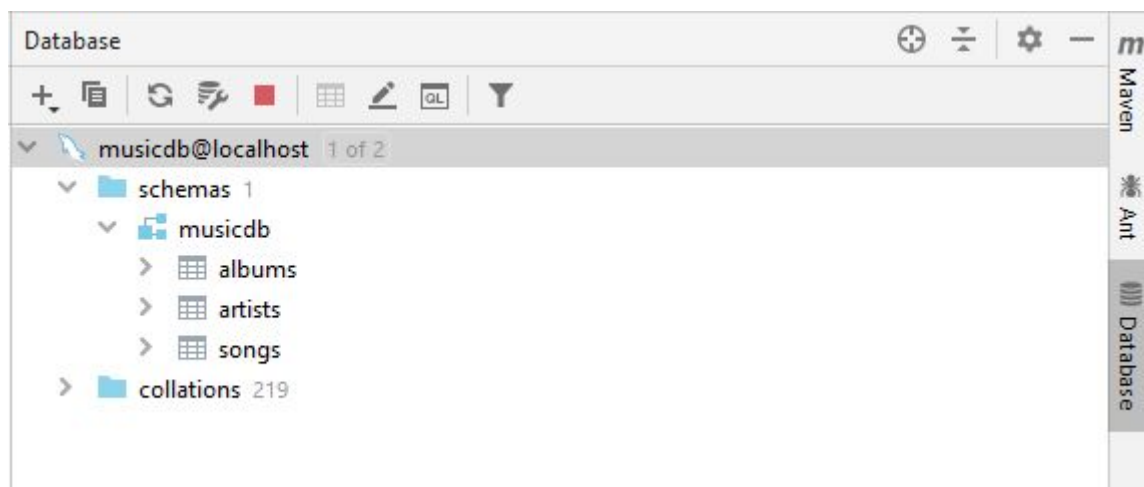
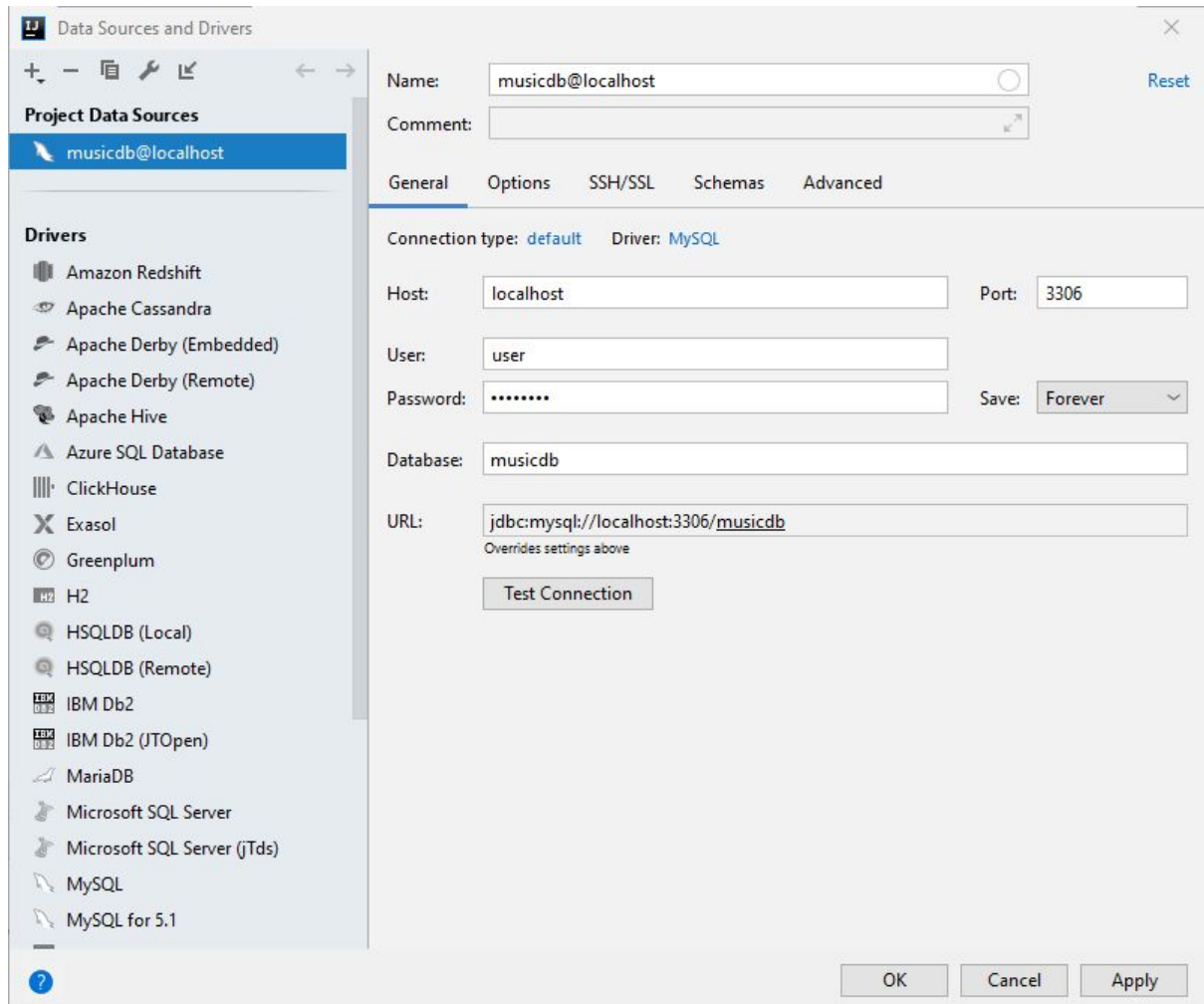


Nadat je de aanpassingen hebt doorgevoerd herstart je docker.

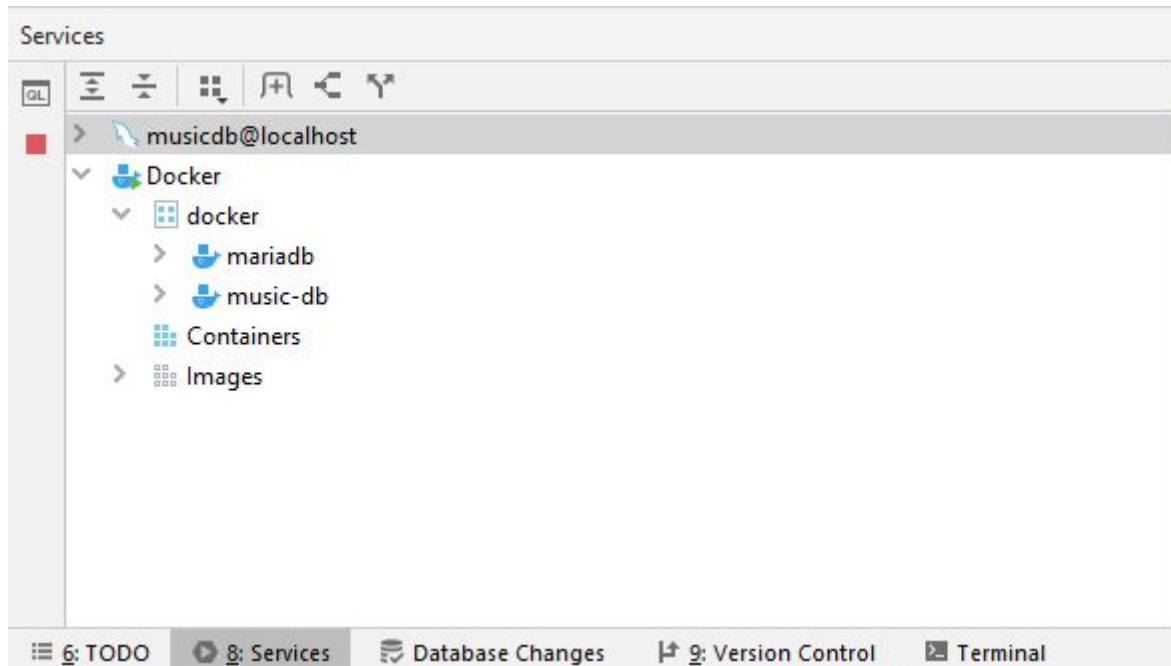
In het bestand `src/main/docker/docker-compose.yml` staat alle informatie om de database op te starten in een docker container. Hiervoor open je een terminal windows (in IntelliJ IDEA), navigeer je naar de directory `src/main/docker` en typ je het commando:

docker-compose up

Zodra de docker container met de mysql database is opgestart kan je via IntelliJ IDEA reeds verbinding maken met de database om te controleren of alle tabellen aanwezig zijn:



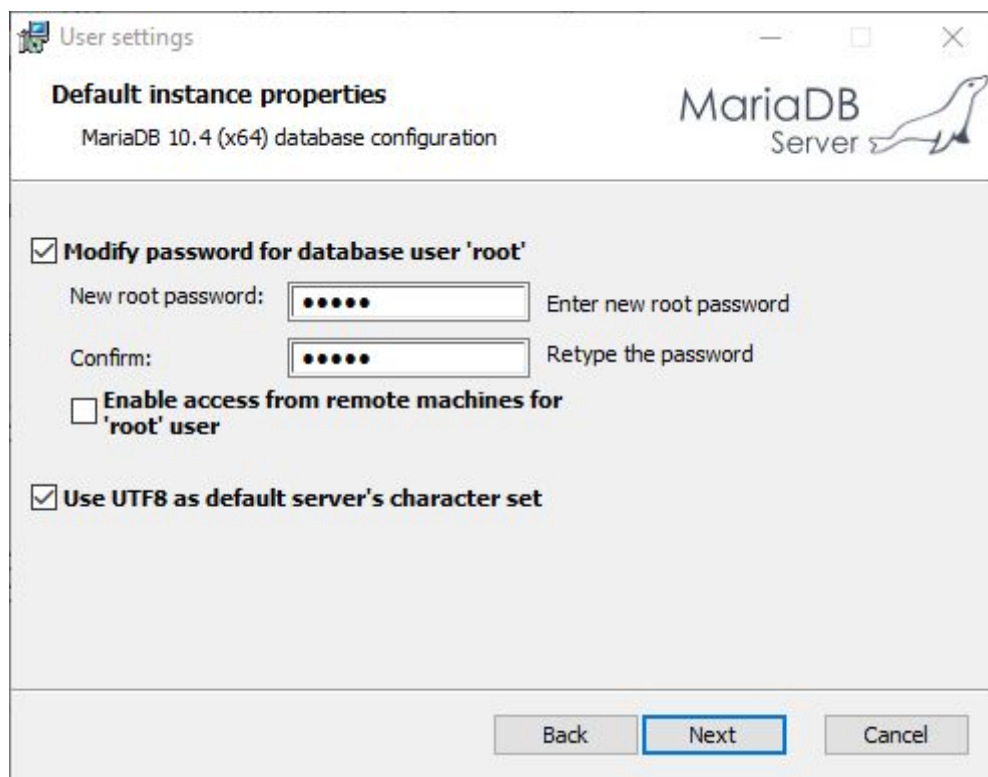
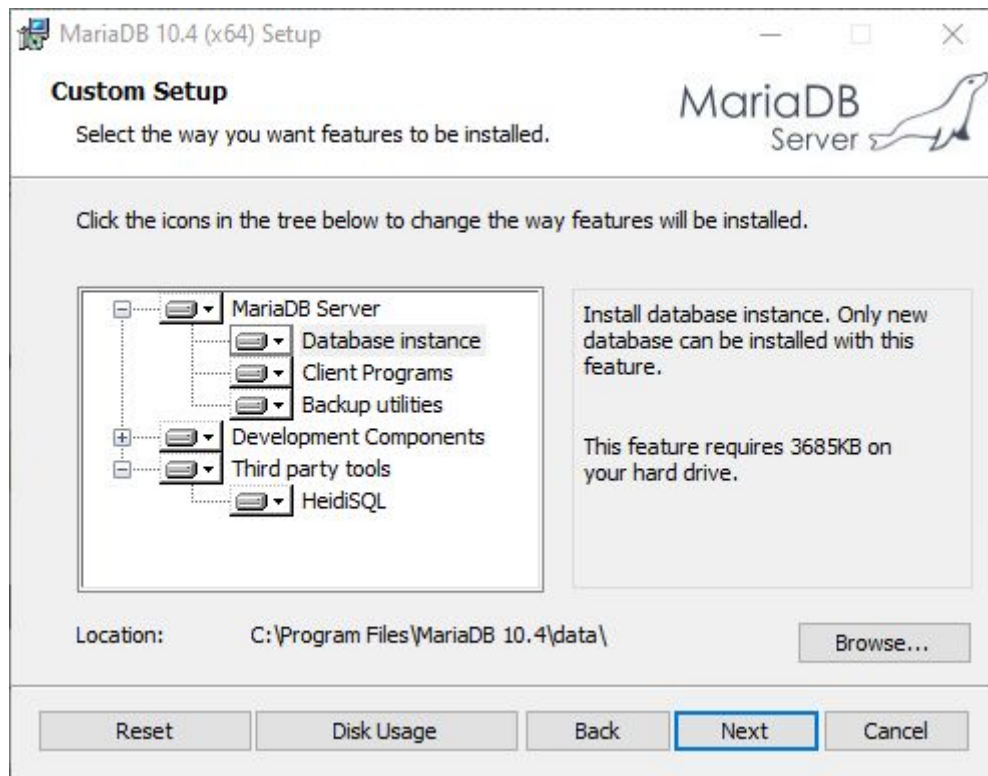
Je kan ook vanuit IntelliJ IDEA de docker containers beheren. Dit vind je terug in het tabblad services:



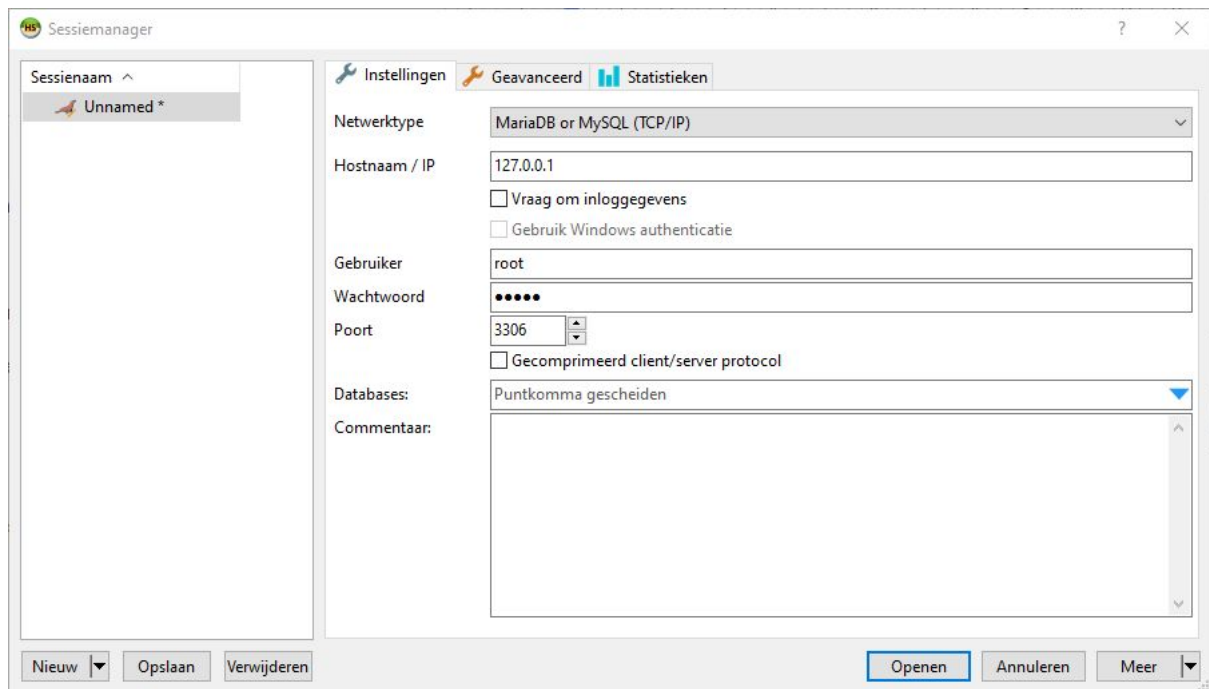
Hier kan je een docker container starten, stoppen en verwijderen.

1.2 Installatie MariaDB

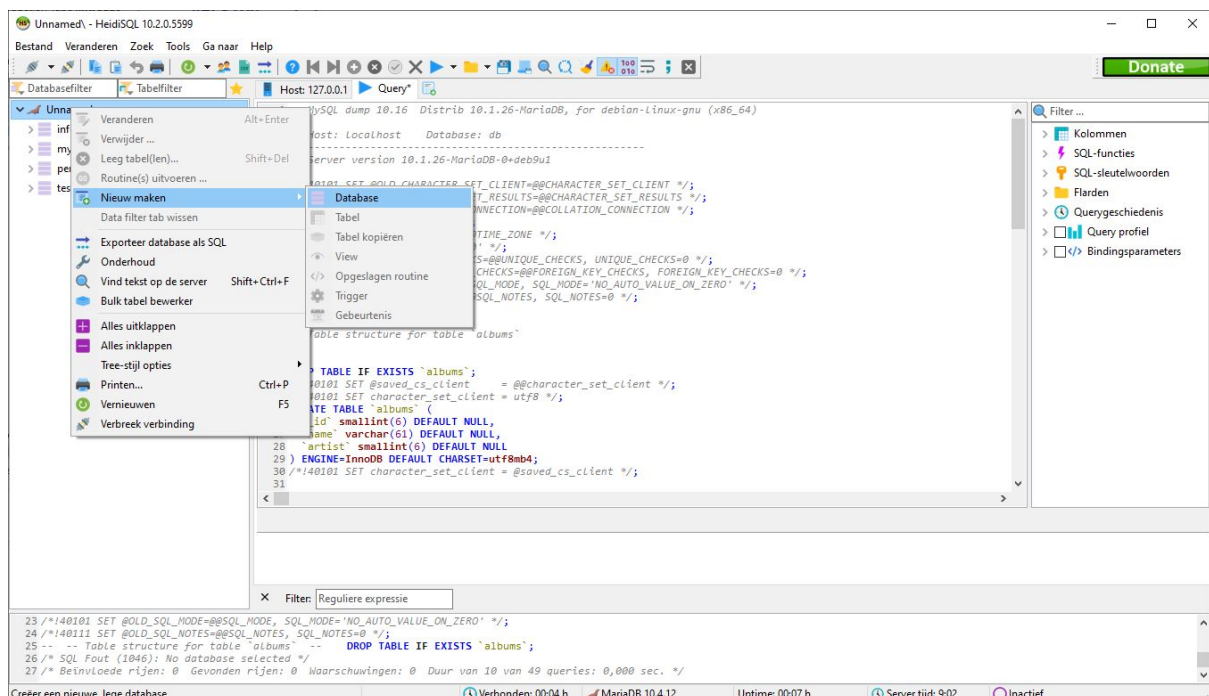
Download de MariaDB server op <https://downloads.mariadb.org/>
Kies voor de huidige stable version (MariaDB 10.4 Series).



De MariaDB service wordt automatisch gestart en met behulp van de tool HeidiSQL kan je de nodige users, tabellen en data aanmaken.



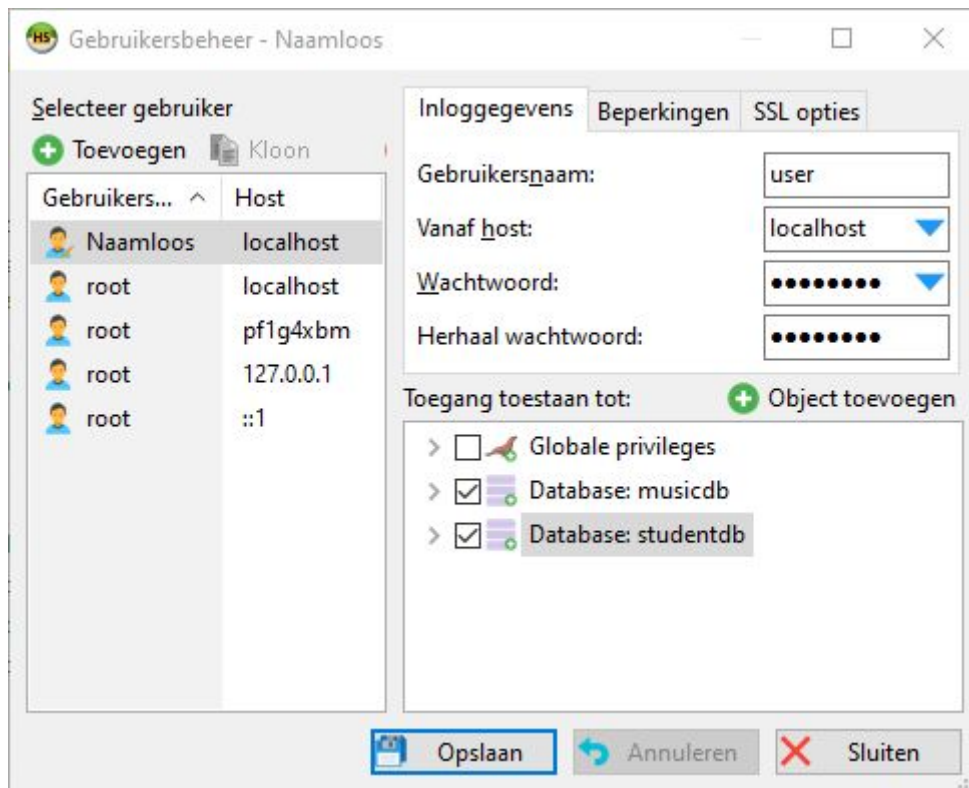
We maken, nadat we verbinding hebben met de databank, 2 databases aan: musicdb en studentdb.



Selecteer eerst de aangemaakte databank musicdb, open het bestand musicdb.sql en voer het bestand uit.

Selecteer vervolgens databank studentdb, open het bestand studentdb.sql en voer het bestand uit.

Maak vervolgens een gebruiker aan die je toegang geeft tot beide databanken.



Zorg vervolgens dat je in de pom.xml van je java project de juiste driver toevoegt:

```
<dependency>
  <groupId>org.mariadb.jdbc</groupId>
  <artifactId>mariadb-java-client</artifactId>
  <version>2.4.4</version>
</dependency>
```

Je kan nu de volgende url gebruiken om met behulp van jdbc verbinding te maken met de databank: **jdbc:mariadb://localhost:3306/musicdb?useSSL=false** of **jdbc:mariadb://localhost:3306/studentdb?useSSL=false**

Opmerking: Indien je bij de installatie een andere poortnummer hebt ingesteld, zal je dit poortnummer in de url moeten aanpassen.

2. Verbinding met de database en SQL commando's gebruiken

Zie handboek:

- 3.1 Inleiding
- 3.2 De databaseserver
- 3.3 Verbinding maken met een database

Demo 1: CreateTable

Verbinding maken en tabel aanmaken

Zie ook handboek 3.4.1 Een statement creëren

Demo 2: InsertUpdateDelete

Zie ook handboek:

3.4.2 Gegevens uit een database opvragen

3.4.3 Gegevens wijzigen in een database

3.4.4 Gegevens toevoegen aan een database

Demo 3: QueryArtists

Demo 4: QueryAlbumsForArtist

Demo 5: CountingRows

Demo 6: QueryForSongInfo

Scenario 1

Enter a title:

Go Your Own Way

Artist name = Fleetwood Mac Album name = The Dance Track = 15

Artist name = Fleetwood Mac Album name = The Very Best Of Track = 1

Artist name = Fleetwood Mac Album name = Greatest Hits Track = 2

Artist name = Fleetwood Mac Album name = Rumours Track = 5

Scenario 2

Enter a title:

Go Your Own Way' or 1=1 or '

Artist name = Mahogany Rush Album name = Mahogany Rush Live Track = 4

Artist name = Mahogany Rush Album name = Mahogany Rush Live Track = 1

Artist name = Mahogany Rush Album name = Mahogany Rush Live Track = 7

Artist name = Mahogany Rush Album name = Mahogany Rush Live Track = 9

Artist name = Mahogany Rush Album name = Mahogany Rush Live Track = 11

Artist name = Mahogany Rush Album name = Mahogany Rush Live Track = 2

Artist name = Mahogany Rush Album name = Mahogany Rush Live Track = 12

Artist name = Mahogany Rush Album name = Mahogany Rush Live Track = 6

....

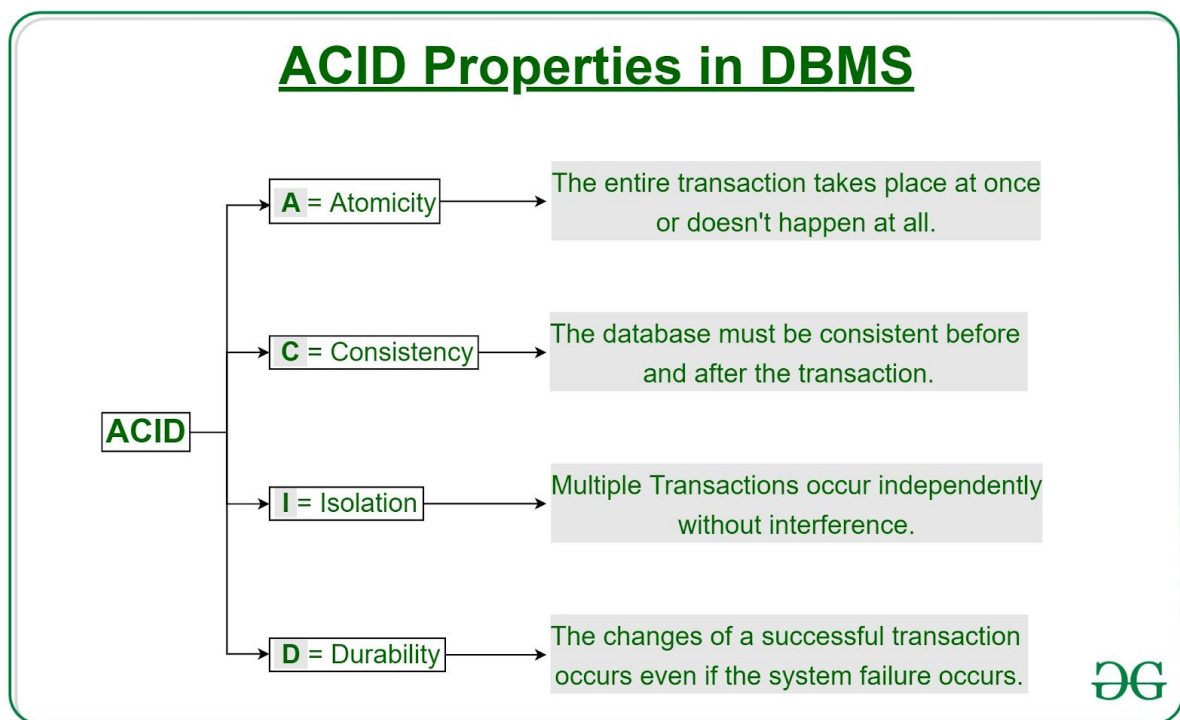
Scenario 2 is een voorbeeld van **SQL injection**.

Dit probleem kunnen we oplossen door gebruik te maken van Prepared statements.

Pas de code aan door gebruik te maken van een prepared statement en test je code uit.

Zie ook handboek: 3.4.5 Prepared statements

3. Transacties



Bron: <https://www.geeksforgeeks.org/acid-properties-in-dbms/>

Bron: <https://www.youtube.com/watch?v=VRm2UMsFVz0>

Afhankelijk van het Transaction Isolation Level kunnen bepaalde scenario's zich voordoen:

- Dirty read
- Non-repeatable read
- Phantom read

Bron: <http://javadata.blogspot.com/2011/05/dirty-readphantom-read-and-non.html>

Zie handboek: 3.5 Transacties

Demo 7: InsertSong

4. DAO (Data Access Objects)

We brengen de code die toegang nodig heeft tot de database onder in aparte klassen. De objecten die de toegang tot de databank verzorgen noemen we Data Access Objects (DAO's).

Demo 8: UsingContactDao

5. Opdrachten

Fork de github repository https://github.com/custersnele/JA2_opdrachten_jdbc.

In de directory src/main/docker vind je een docker-compose.yml waarmee je een docker-container met MariaDB databank kan opstarten met de tabellen voor de opdrachten. De URL om met deze databank te connecteren is:

`jdbc:mariadb://localhost:3307/studentdb?useSSL=false`

Je kan ook gebruik maken van je lokaal geïnstalleerde MariaDB.

De opdrachten vind je terug in het handboek.

- Opdracht 2: Gegevens opvragen
- Opdracht 3: Gegevens wijzigen
- Opdracht 4: Gegevens toevoegen aan de database
- Opdracht 5: Een PreparedStatement gebruiken
- Bestudeer de code BankTransfer (transacties)
- Opdracht 8: Data Access Objects