Crie um diretório para você:  


mkdir visualops-files

Crie um docker-compose.yml e uma pasta para armazenar os scripts sql:  




touch docker-compose.yml

mkdir sql-files

Clone o repositório shared\_script e mova o seu jar individual para a pasta que você criou anteriormente:  


git clone <https://github.com/Organizacao-Grupo5/shared_script.git>  
 ou

git clone https://SEU\_TOKEN\_DE\_ACESSO\_DO\_GITHUB@[github.com/Organizacao-Grupo5/shared\_script.git](https://github.com/Organizacao-Grupo5/shared_script.git)

link para criar token = https://github.com/settings/tokens

Deve ficar mais ou menos assim:  
Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Mude para o diretório de scripts sql e crie um arquivo para criar o usuário e as tabelas:  


cd sql-files/

touch create-script.sql

Abra o arquivo com o Vim ou Nano (recomendo Vim pois não sei usar o Nano):  
Texto

Descrição gerada automaticamente

-- SQLBook: Code

DROP DATABASE IF EXISTS visualops\_db;

CREATE DATABASE visualops\_db;

CREATE USER IF NOT EXISTS 'client'@'%' IDENTIFIED BY 'Client123$';

GRANT SELECT, UPDATE, DELETE, INSERT ON visualops\_db.\* TO 'client';

FLUSH privileges;

USE visualops\_db;

use essa configuração antes de criar as tabelas  
  
Cole seu script de banco de dados:  
Texto

Descrição gerada automaticamente

exemplo do meu. (o meu está com algumas alterações que não tem no em grupo pois isso não mandei o script inteiro)

Observação: em tabelas que tem foreign Keys compostas deve-se colocar tudo dentro de uma constraint como mostrado na imagem abaixo:

Texto

Descrição gerada automaticamente

Ao finalizar as mudanças no sql, volte para a pasta que você criou:  
Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Agora mova seu jar individual para a pasta atual:  
  
mv ‘/shared\_scripts/Jar Individuais/jar-individual-SEU\_NOME’ .

Agora já está nesse jeito:  


Em seguida entre no seu jar e crie o Dockerfile:  
Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

cd jar-individual-SEU\_NOME

vim Dockerfile

FROM maven:3.8.5-openjdk-17 AS builder

WORKDIR /app

COPY . /app

RUN mvn clean install

FROM openjdk:23-ea-17-jdk

COPY --from=builder /app/target/jar-individual-thiago-1.0-SNAPSHOT-jar-with-dependencies.jar /app.jar

CMD ["java", "-jar", "/app.jar"]

Agora volte para o seu diretório, mude para o mysql-config e altere o endereço :  
Texto

Descrição gerada automaticamente

vim src/main/java/útil/database/MySQLConnection.java

Depois volte para o seu diretório e edite o Docker-compose.yml. vim docker-compose.yml:  
Texto

Descrição gerada automaticamente

version: '3.3'

services:

mysql:

container\_name: db-container

image: mysql:latest

restart: always

environment:

MYSQL\_ROOT\_PASSWORD: urubu100

MYSQL\_USER: client

MYSQL\_PASSWORD: Client123$

volumes:

- ./sql-files:/docker-entrypoint-initdb.d

ports:

- "3306:3306"

networks:

- connection-mysqljava

java:

container\_name: java-container

build:

context: ./jar-individual-SEU\_NOME/

dockerfile: Dockerfile

restart: always

stdin\_open: true

tty: true

environment:

- DB\_URL=jdbc:mysql://mysql:3306/SEU\_DATABASE

ports:

- "8090:8080"

depends\_on:

- mysql

networks:

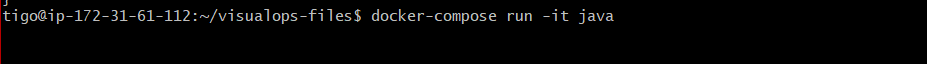
- connection-mysqljava

volumes:

mysql\_data:

networks:

connection-mysqljava:

Volte novamente para seu diretório rode o container java:  


Agora Aprecie!!

Para mais informações ou erros chame no zap.