**Один из способов подключения к базе данных MySQL**

1. **Установить MySQL**

Инструкция по установке: <https://metanit.com/sql/mysql/1.1.php>

1. **Создать проект в IntelliJ IDEA**

Для сборки проекта можно использовать Maven вместо стоящего по умолчанию IntelliJ (в данной инструкции далее будет использоваться Maven). Пример настроек проекта приведен на рис.1.

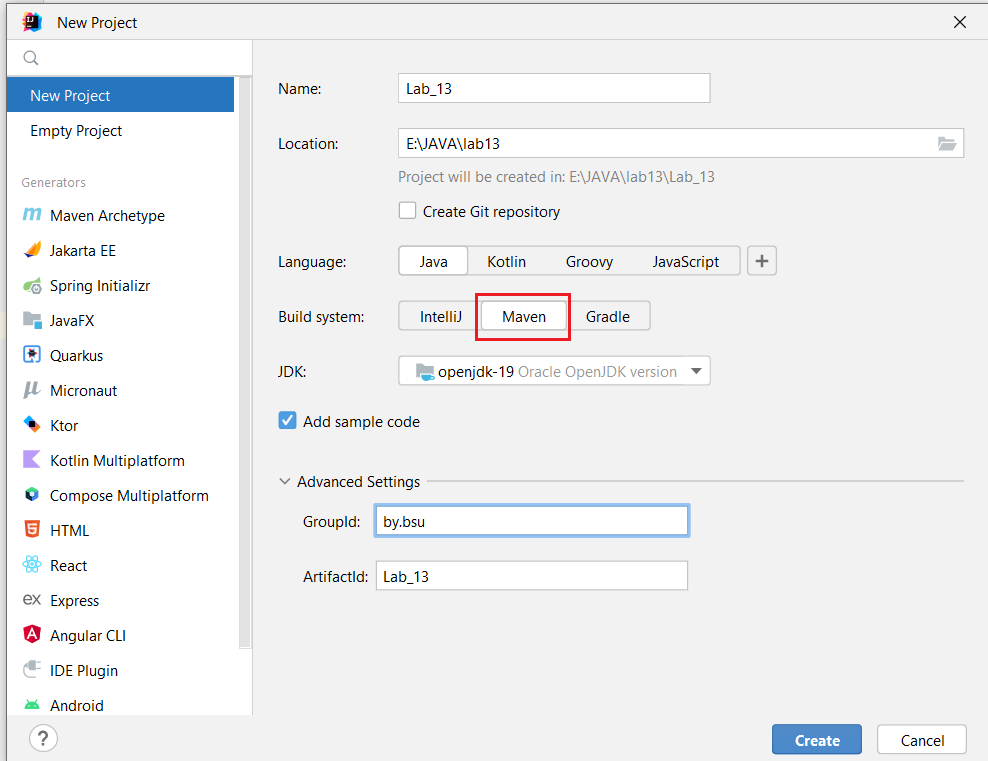


рис.1

1. **В файл pom.xml добавить следующую зависимость:**

**<dependencies>**

**<dependency>**

**<groupId>mysql</groupId>**

**<artifactId>mysql-connector-java</artifactId>**

**<version>8.0.32</version>**

**</dependency>**

**</dependencies>**

Версию можно заменить на любую из предложенных. Если зависимость не установилась и подсвечивается красным, то нажать на появившуюся кнопку в правом верхнем углу (рис.2)

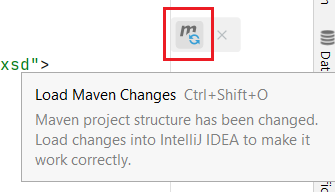


рис.2

1. **Откроем ранее установленный MySQL Workbench (см. пункт 1)**

Если установка и настройка пользователя прошла правильно, то при открытии увидим следующее (рис.3)

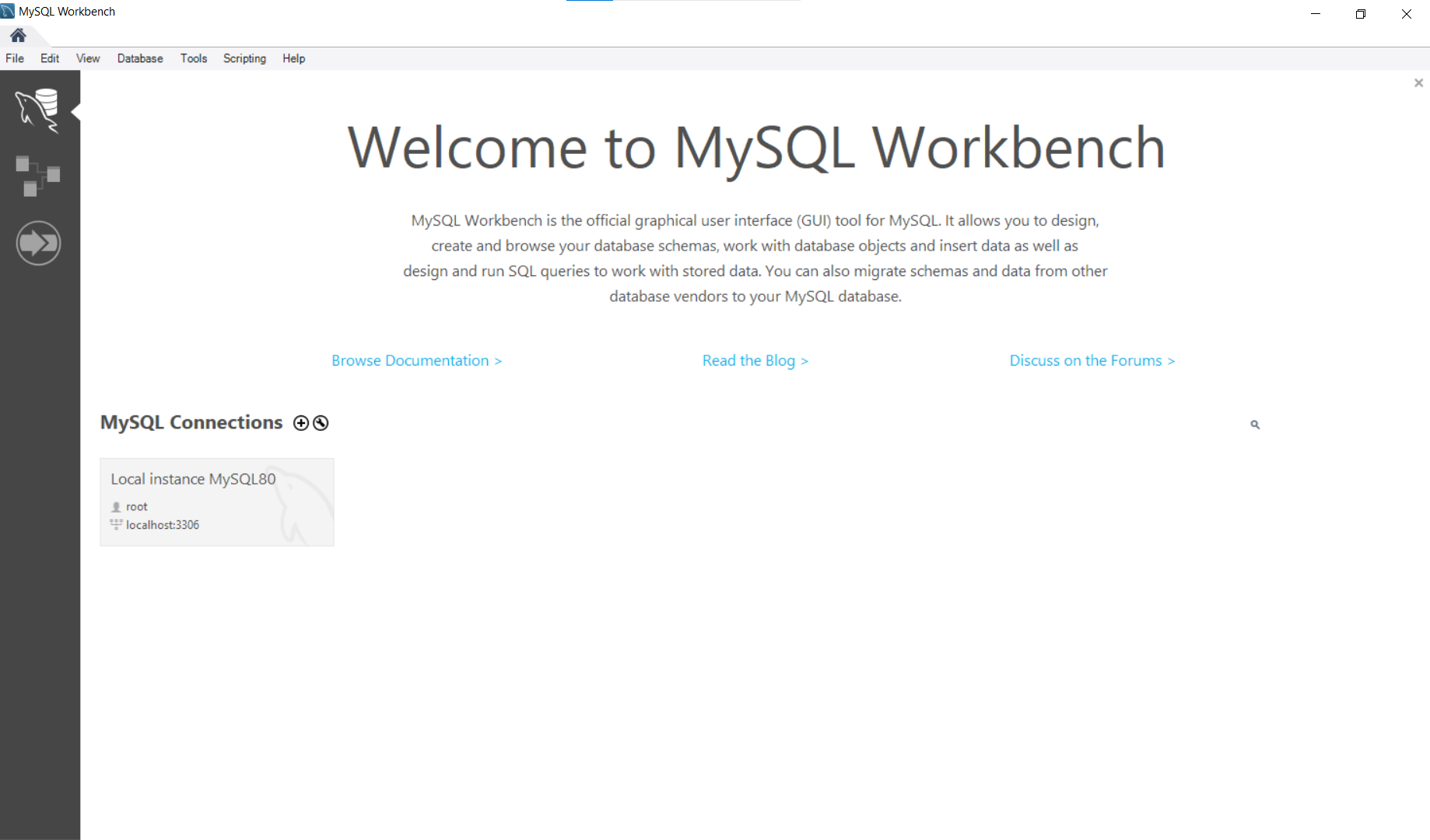


рис.3

1. **Теперь есть 2 варианта создания базы данных: через создание новой схемы и вариант создания через консоль. Рассмотрим вариант через консоль (вариант со схемой выполняется аналогичным образом, только сразу в Local instance MySQL создается новая схема с названием mails)**
2. правой кнопкой мыши нажать на Local instance MySQL и выбрать Start Command Line Client (рис.4)

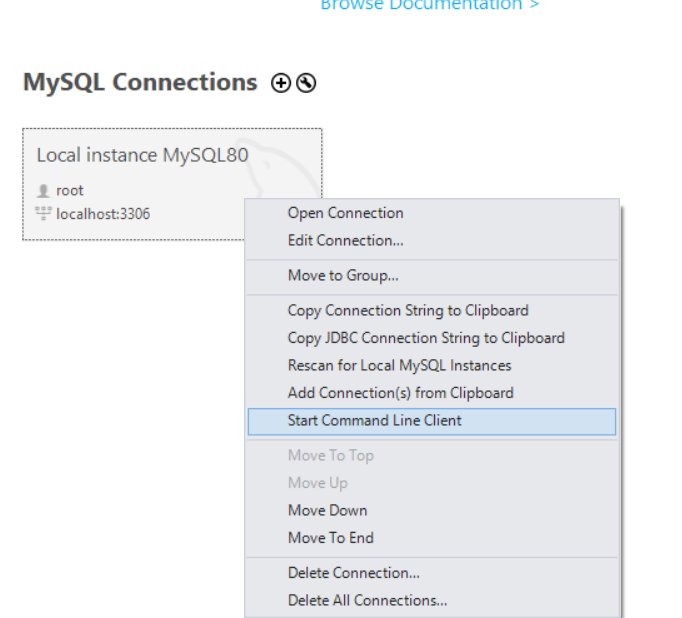


рис.4

1. в открывшемся окне ввести пароль (тот, который вы вводили, когда создавали пользователя). В моем случае пользователь root имеет пароль $Jn4B42hKNj44Z@d

*Примечание:* пароль нельзя скопировать и вставить в консоль, поэтому нужно быть внимательным при его вводе, иначе появится ошибка (рис.5)

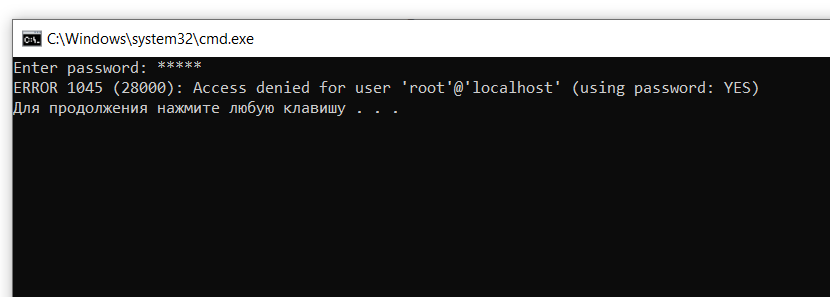


рис.5

Если все выполнено правильно, то увидим рис.6

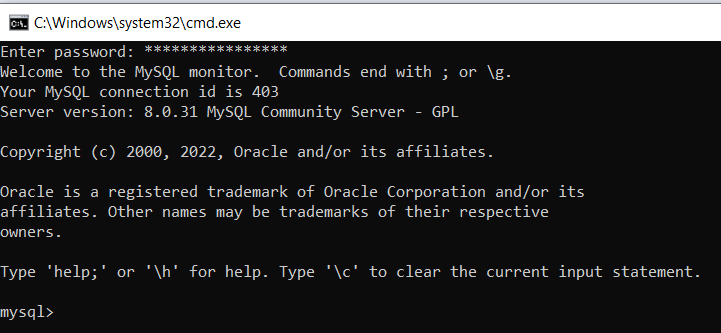


рис.6

1. теперь просмотрим уже существующие базы данных с помощью следующей команды:

**mysql> show databases;**

*Примечание:* после каждой команды необходимо ставить точку с запятой (;) иначе он не будет отрабатывать и в итоге приведет к ошибке.

Примерно такой список должен будет вывестись на консоль, если команда написана правильно (рис.7)

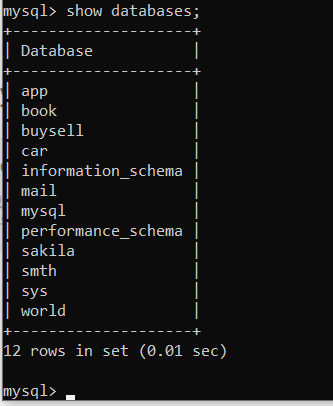


рис.7

Здесь можно просмотреть, какие базы данных есть по умолчанию, а какие, возможно, когда-то уже были созданы вами

1. создадим базу данных (у каждого варианта она будет своя, но для примера возьмем БД письма (mails) (рис.8)

**mysql> create database mails;**

****

рис.8

*Примечание:* если такая бд уже существует и вы хотите ее удалить, то можно использовать команду:

**mysql> drop database mails;**

1. после создания базы данных можно закрыть консоль и открыть Local instance MySQL в Workbench (рис.9)

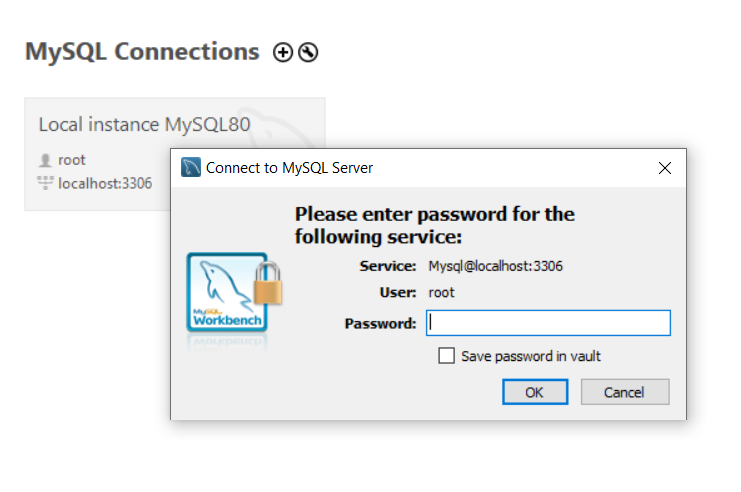


рис.9

Здесь снова потребуется ввести пароль. После входа произойдет переход на страницу со схемами (рис.10)

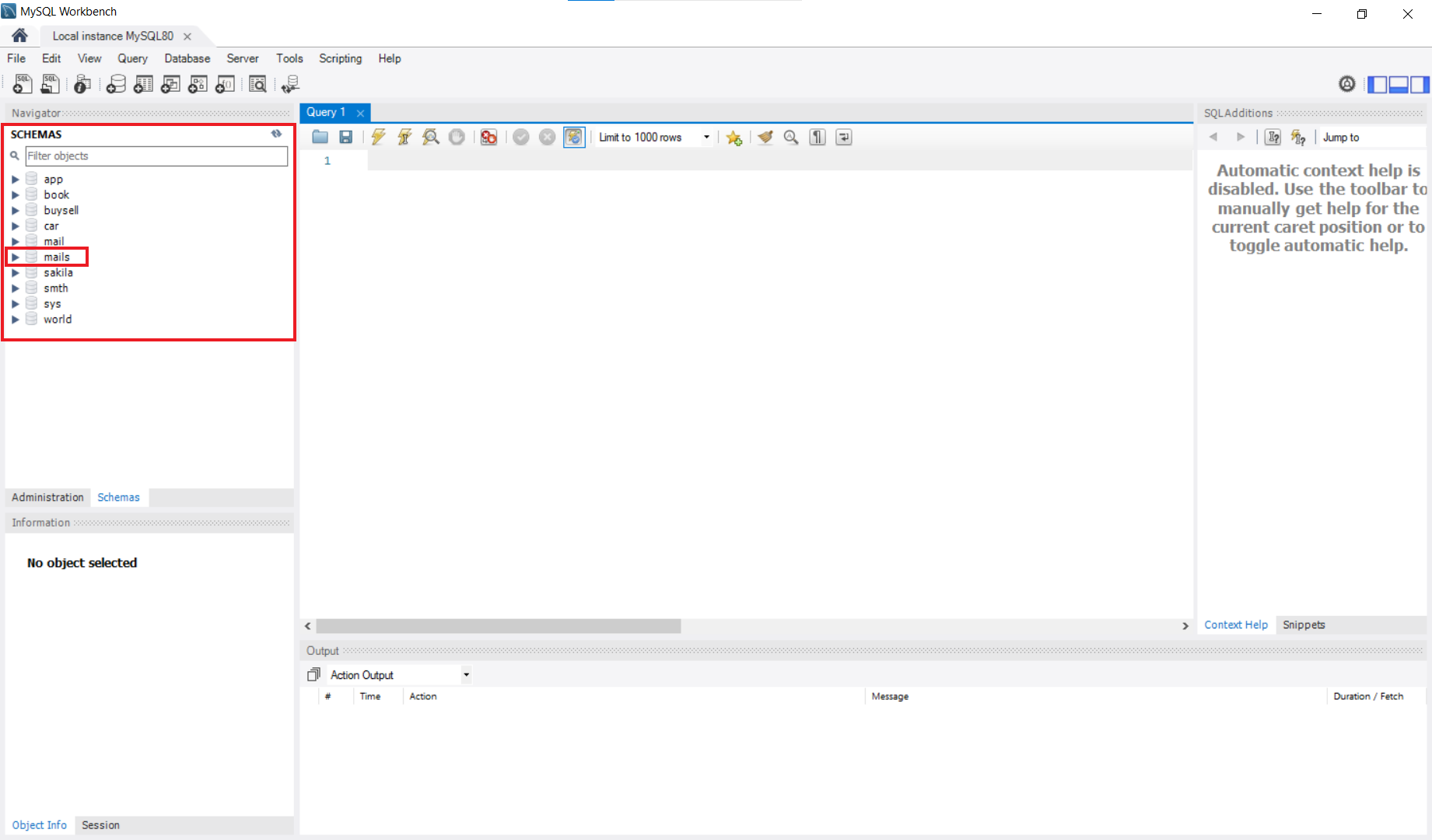


рис.10

1. теперь нужно создать таблицы БД и соответственно их заполнить.

Для начала определимся, какие данные будут храниться в создаваемой базе данных. Например, для писем это будут “mails” (письма) и “people” (пользователи). Эти 2 таблицы и будем создавать и заполнять.

Развернув схему mails и нажав правой кнопкой мыши на таблицы (Tables), выберем Create Table (рис.11)

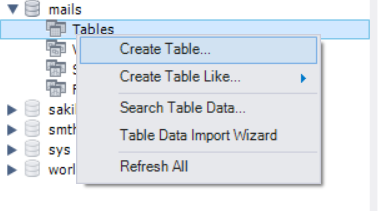


рис.11

1. дадим название таблице people и добавим имена объектов бд (id как PK, fio - ФИО человека, date\_birth - его дата рождения). Определим типы данных и настройки. После завершения нажмем Apply (рис.12)

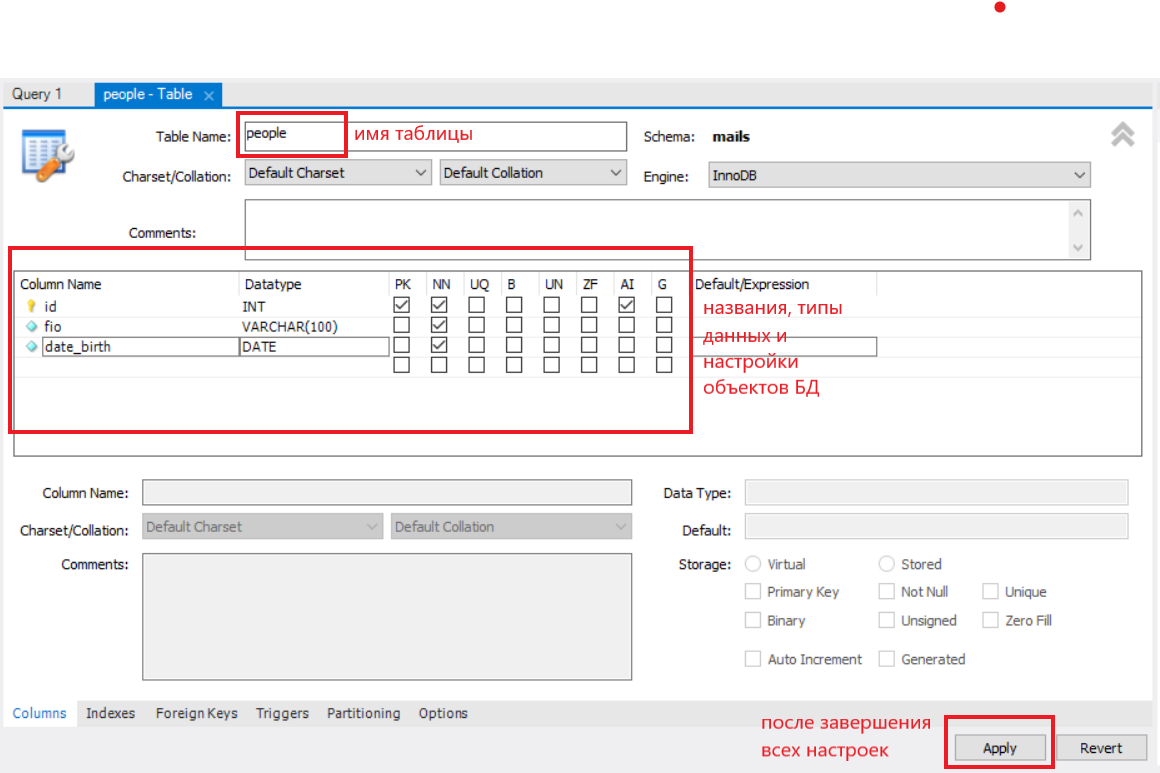


рис.12

В окне (рис.13) перепроверить, все ли введено правильно и еще раз нажать Apply, а затем Finish (рис.14)

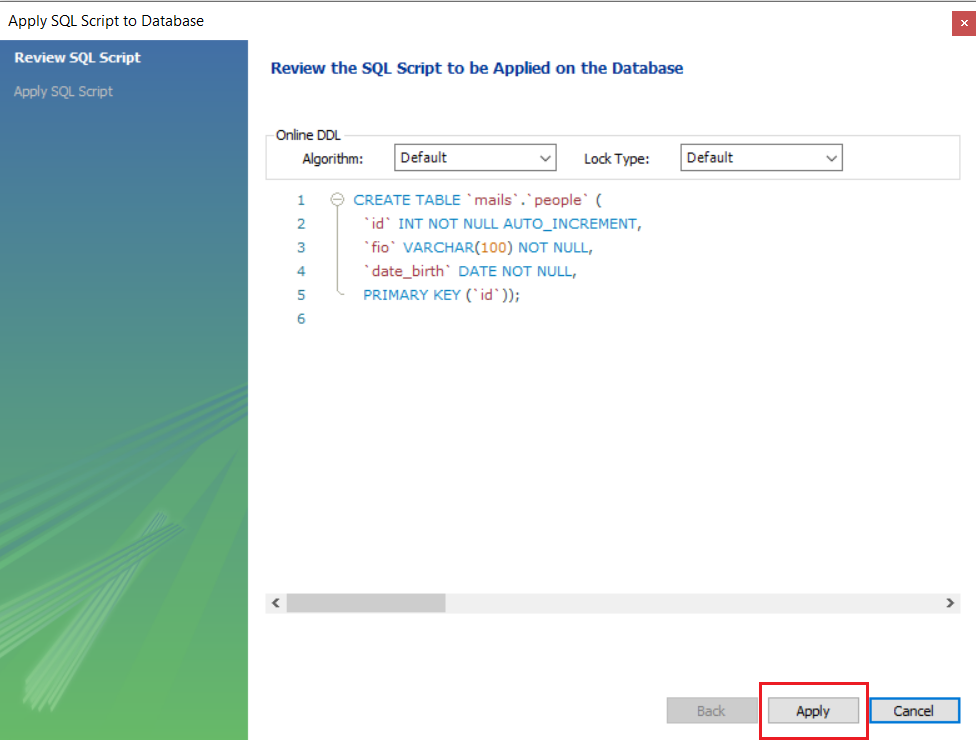


рис.13

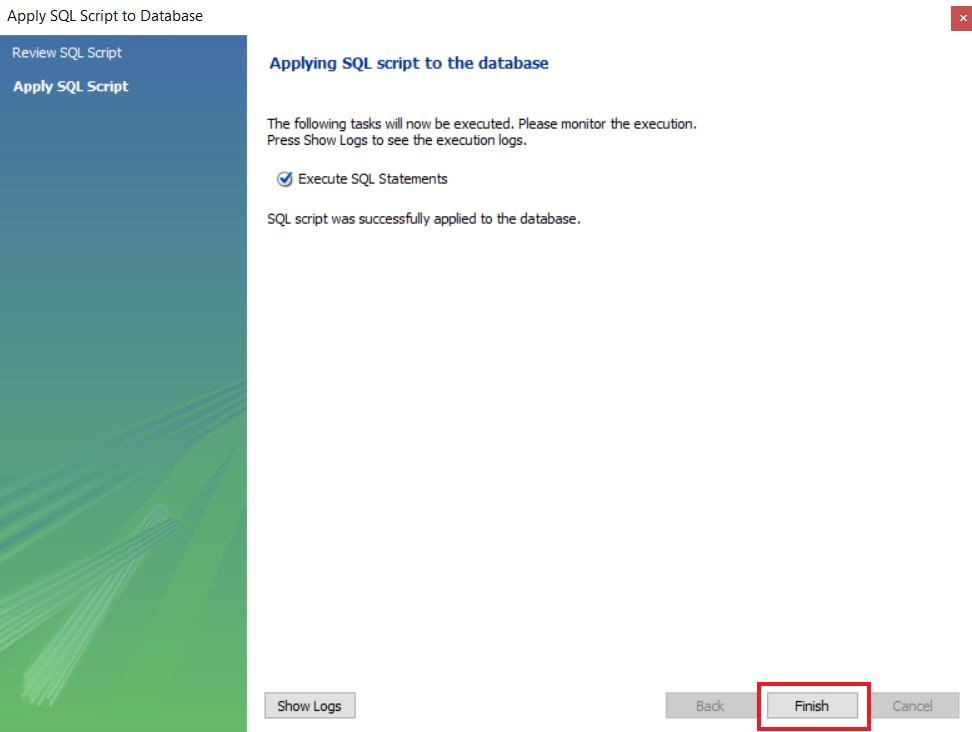


рис.14

1. аналогично создать вторую таблицу mails (рис.15), **НО** не нажимать пока Apply

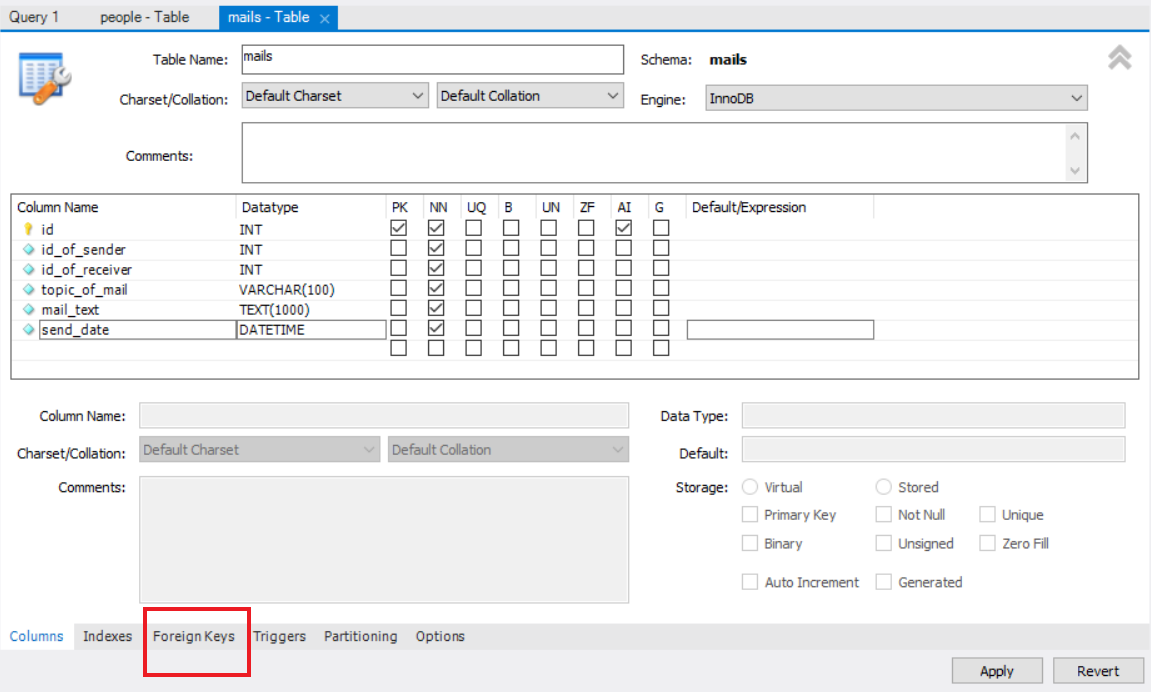


рис.15

Перейдем на вкладку Foreign Keys (см. красный прямоугольник на рис. 15). Добавим имена foreign key’ям и определим, к каким таблицам и элементам они будут относиться (рис.16, рис.17)

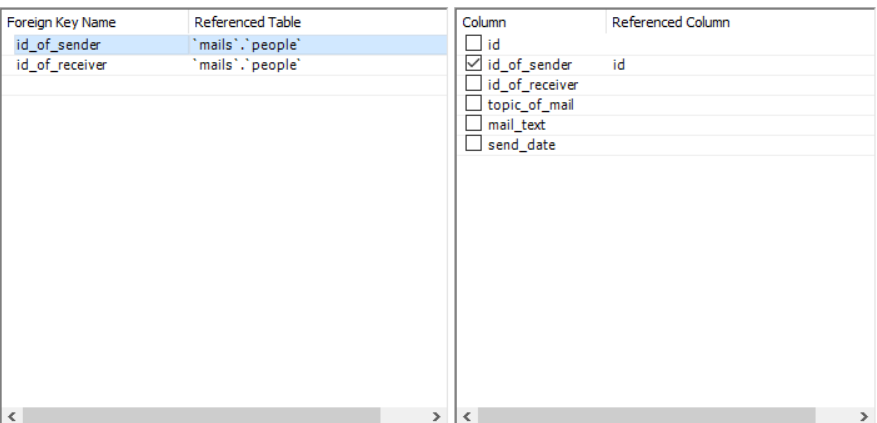


рис.16

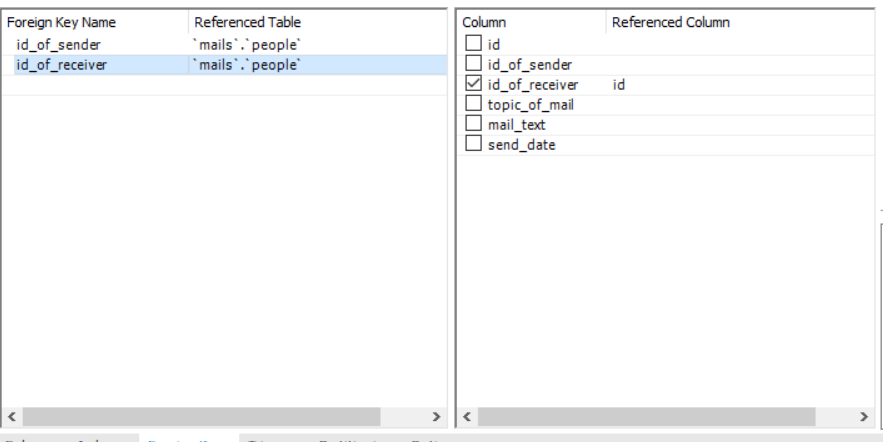


рис.17

Теперь можно нажимать на Apply. В окне выбираем еще раз Apply (рис.18), потом Finish и проверяем, чтобы все было выполнено без ошибок в Output, который находится в самом низу вкладки (рис.19)

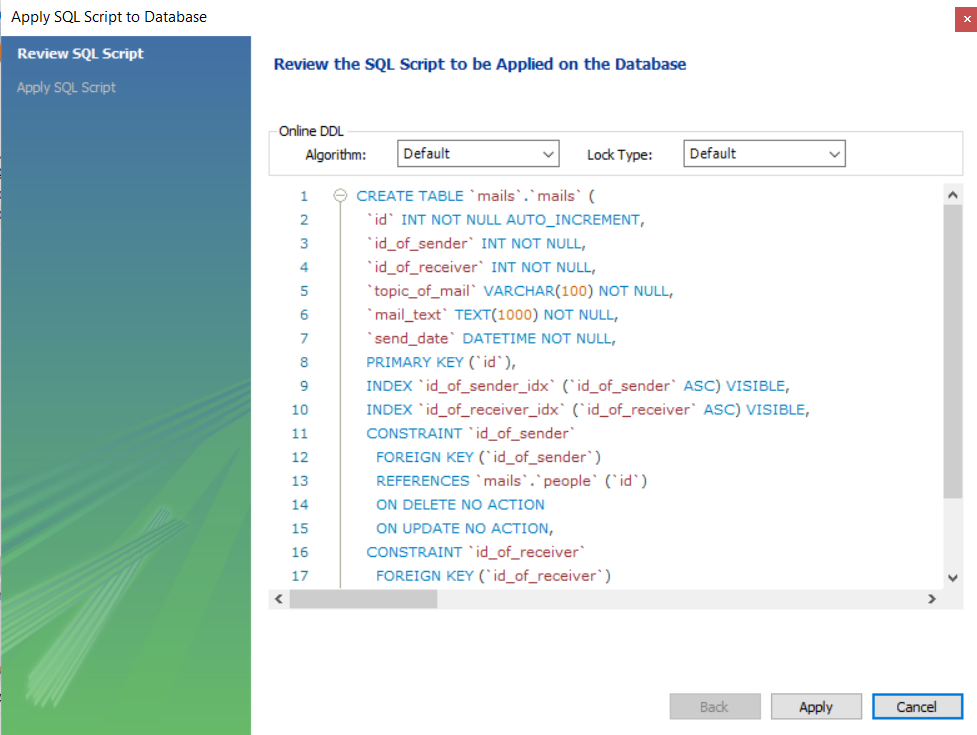


рис.18

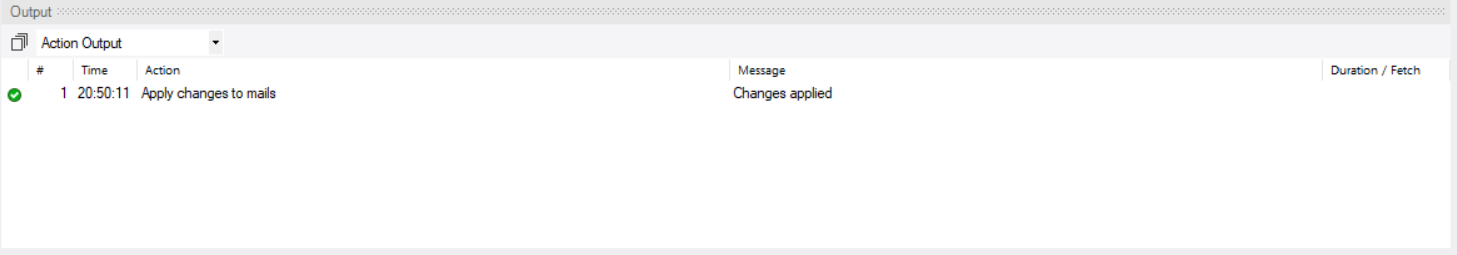


рис.19

1. теперь можно закрыть Workbench и вернуться в IntelliJ IDEA в свой проект

**6. Создадим класс DBConnector, который будет отвечать за подключение БД (рис.20). Код представлен ниже**

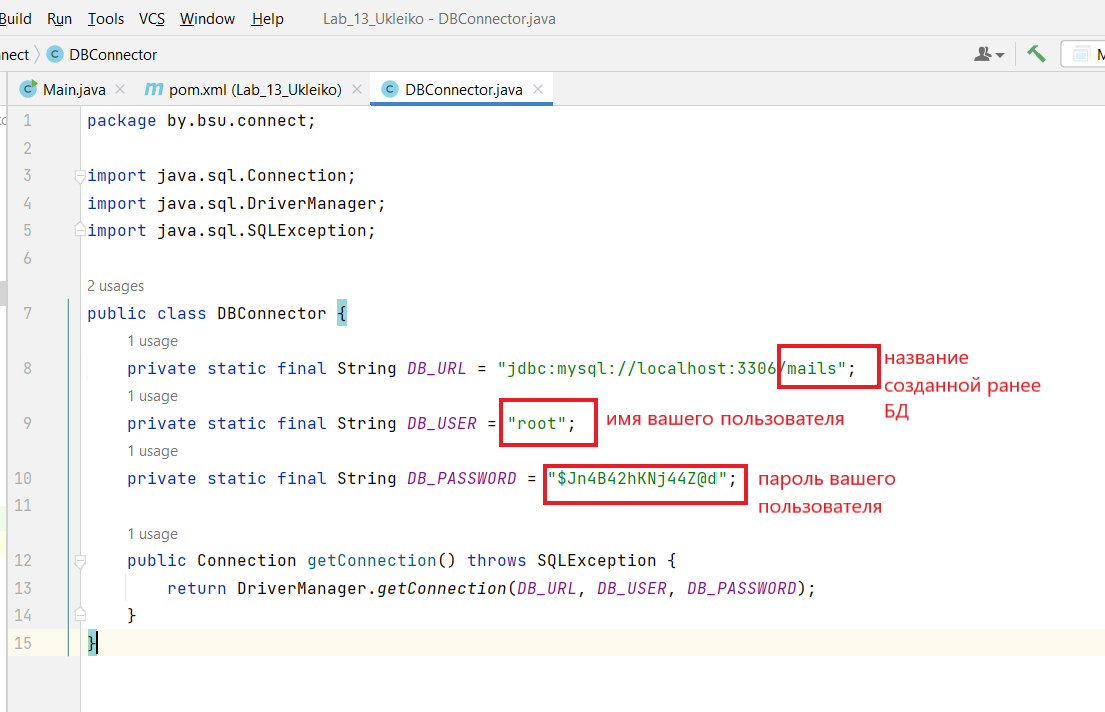
****

рис.20

**package by.bsu.connect;**

**import java.sql.Connection;**

**import java.sql.DriverManager;**

**import java.sql.SQLException;**

**public class DBConnector {**

**private static final String *DB\_URL* = "jdbc:mysql://localhost:3306/mails";**

**private static final String *DB\_USER* = "root";**

**private static final String *DB\_PASSWORD* = "$Jn4B42hKNj44Z@d";**

**public Connection getConnection() throws SQLException {**

**return DriverManager.*getConnection*(*DB\_URL*, *DB\_USER*, *DB\_PASSWORD*);**

**}**

**}**

**7. База данных подключена. Осталось только создать класс для добавления в нее строк, а также написать запросы к БД.**

**8. В Main прописать следующее:**

**public class Main {**

**public static void main(String[] args) {**

**try (Connection connection = new DBConnector().getConnection()) {**

**//код в соответствии с заданием**

**} catch (SQLException e) {**

**e.printStackTrace();**

**}**

**}**

**}**