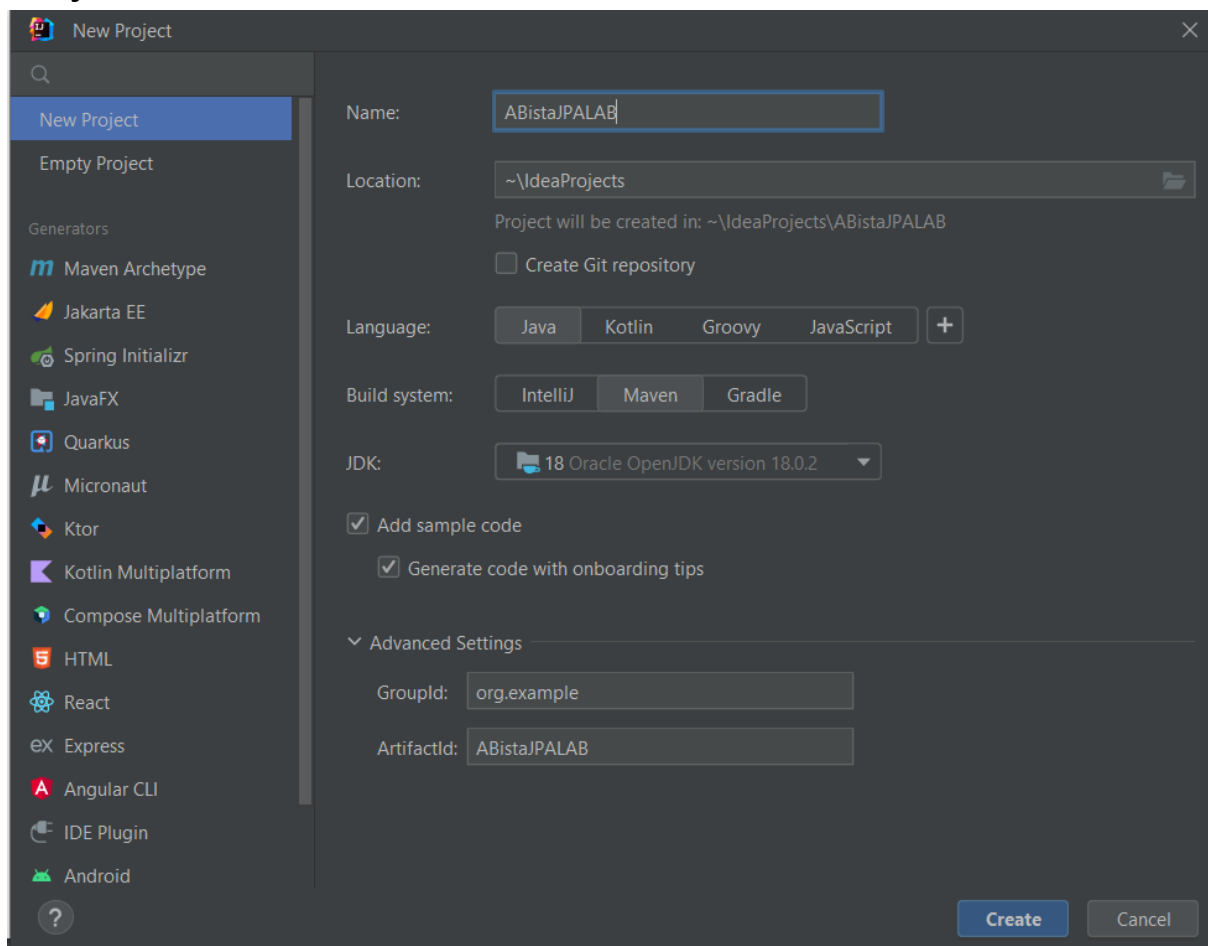


Adam Biśta

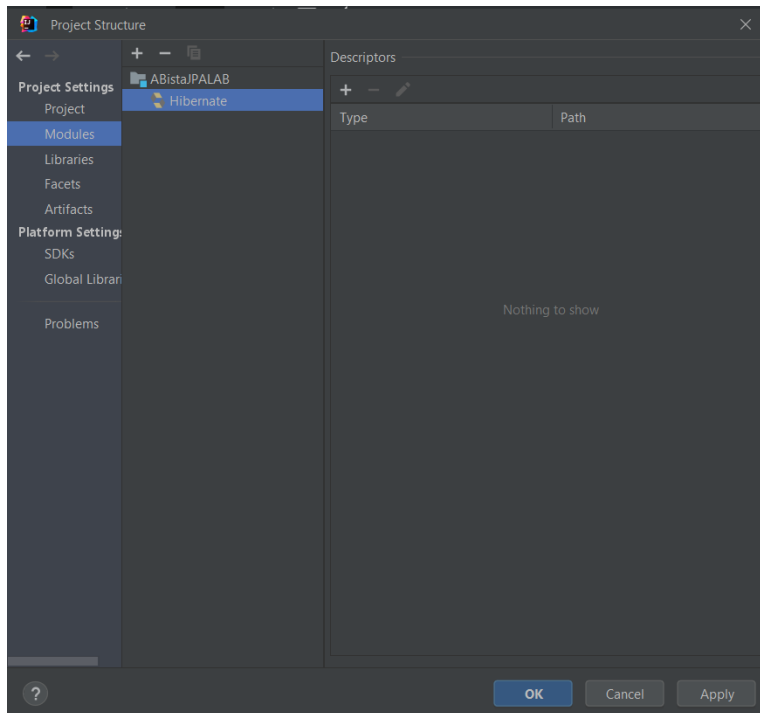
1. Załączamy serwer Derby

```
C:\WINDOWS\system32\cmd.exe
Thu Apr 13 11:39:46 CEST 2023 : Serwer sieciowy Apache Derby - 10.16.1.1 - (1901046) uruchomiony i gotowy do zaakceptowa
nia połączeń na porcie 1527 w {3}
```

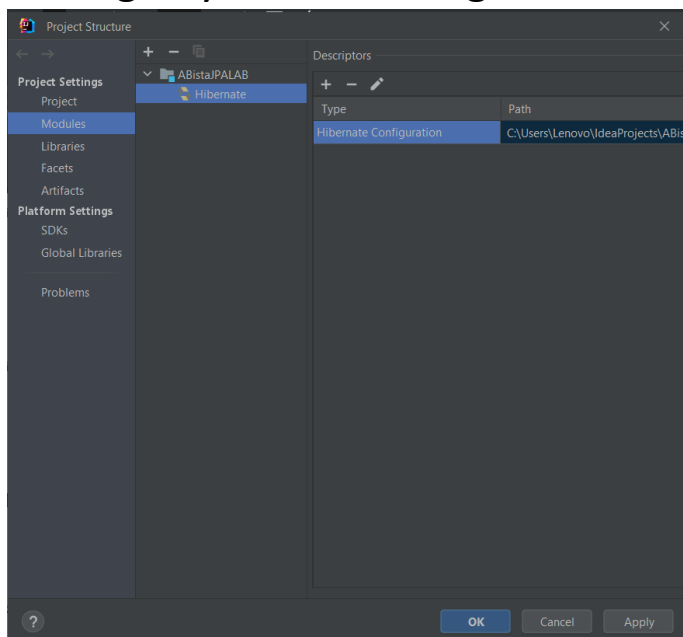
2. Projekt w IntelliJ



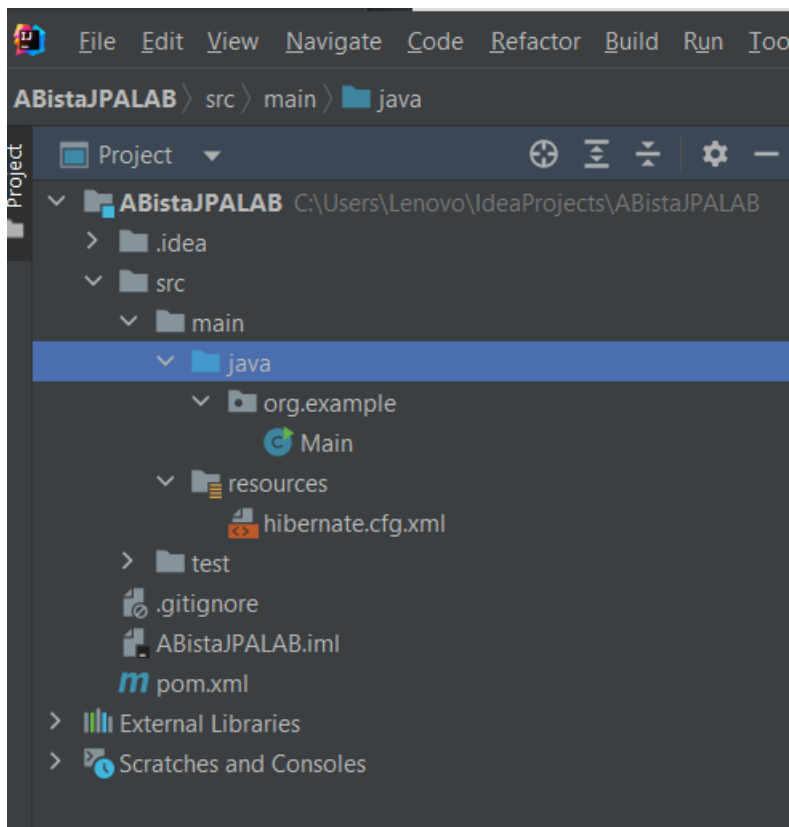
3. Dodajemy Hibernate:



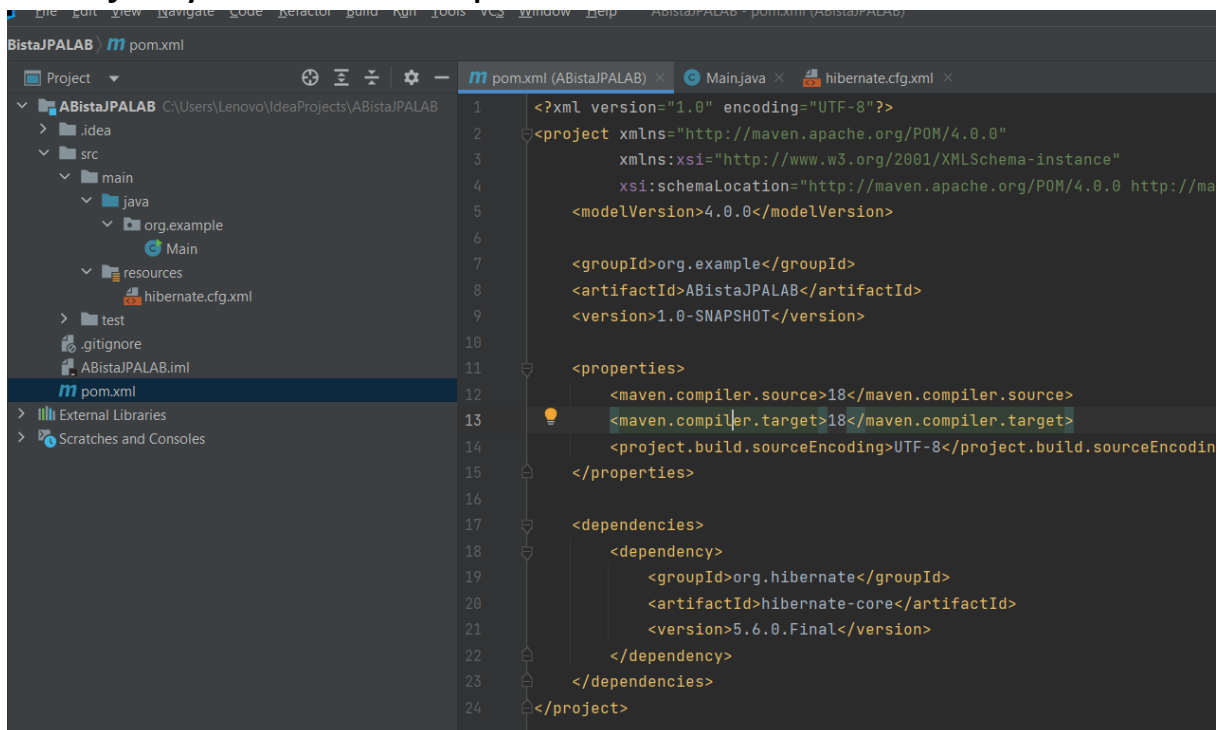
4. Dociągamy hibernate.cfg.xml:



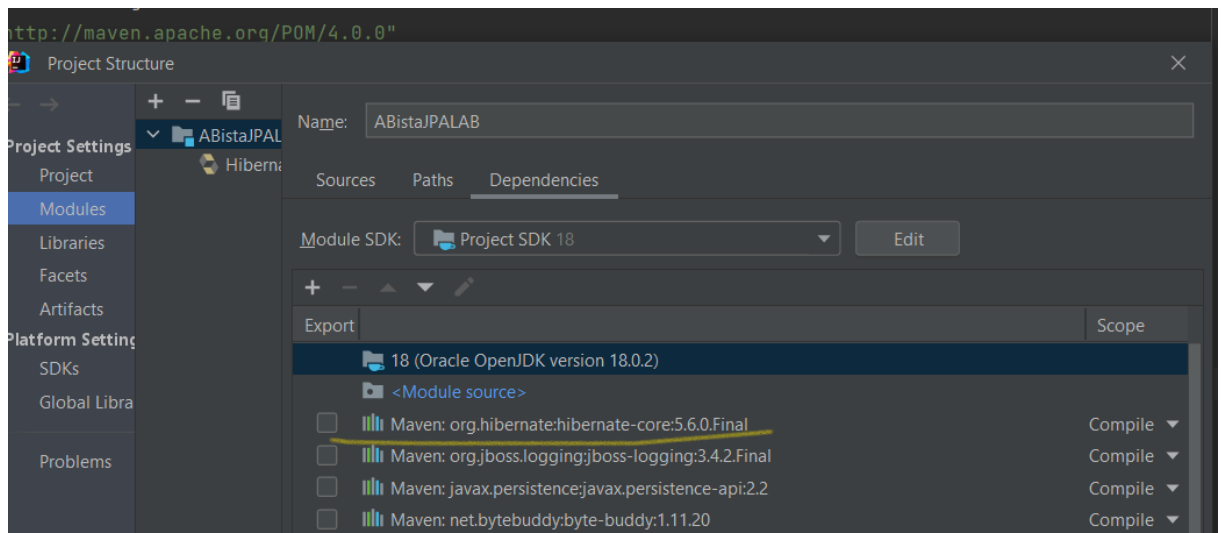
5. Aktualny widok w strukturze projektu:



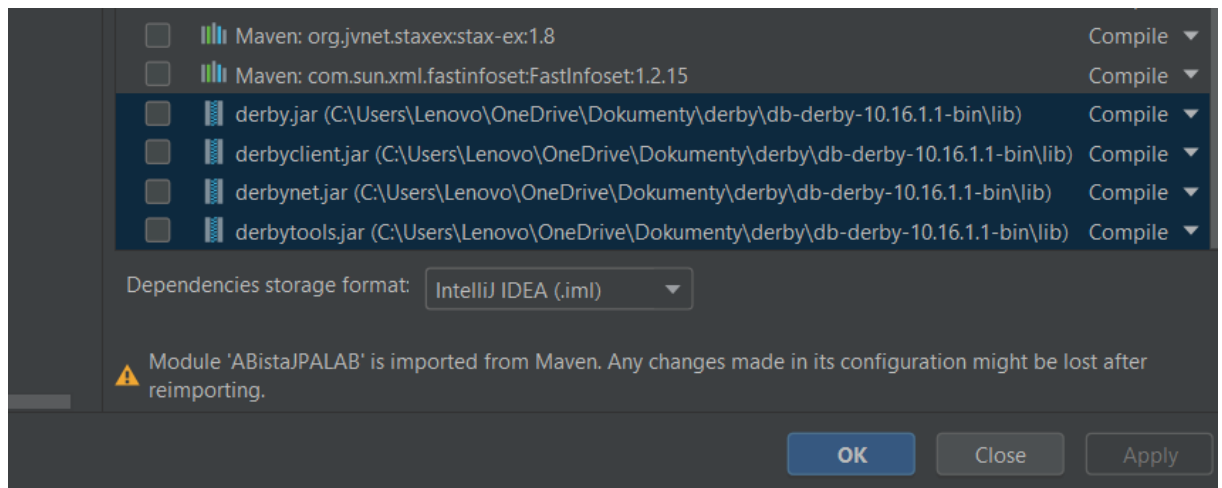
6. Dodajemy zależność do pom.xml:



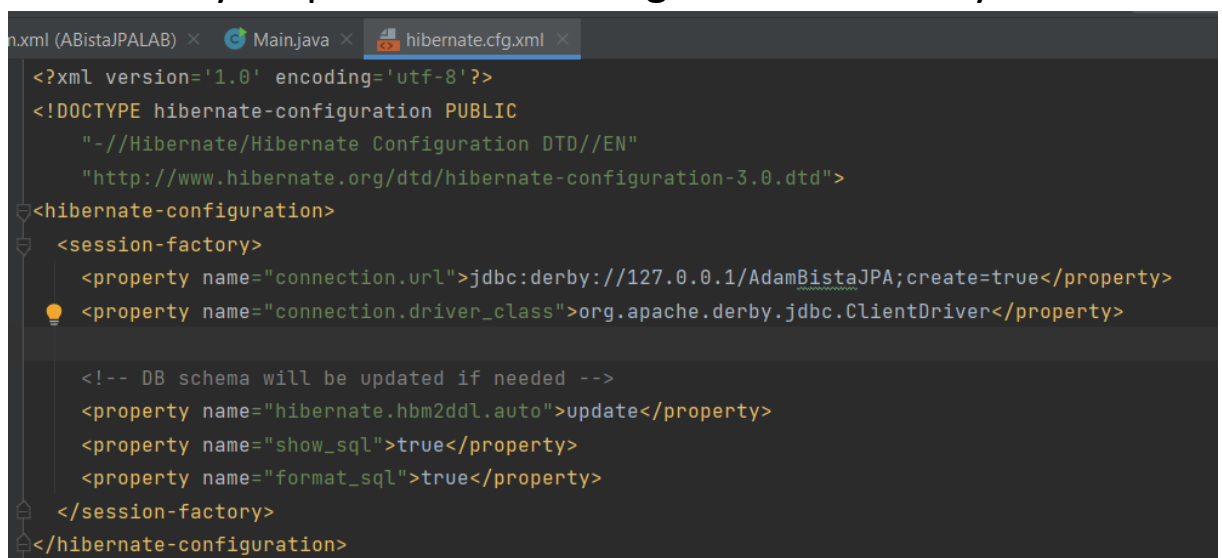
7. Hibernate w zależnościach:



8. Dociągamy Jarki:



9. Wchodzimy do pliku hibernate.cfg.xml i ustawiamy:



10. Klasa Main:

11.

```
package org.example;

import org.hibernate.HibernateException;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class Main {
    private static final SessionFactory ourSessionFactory;

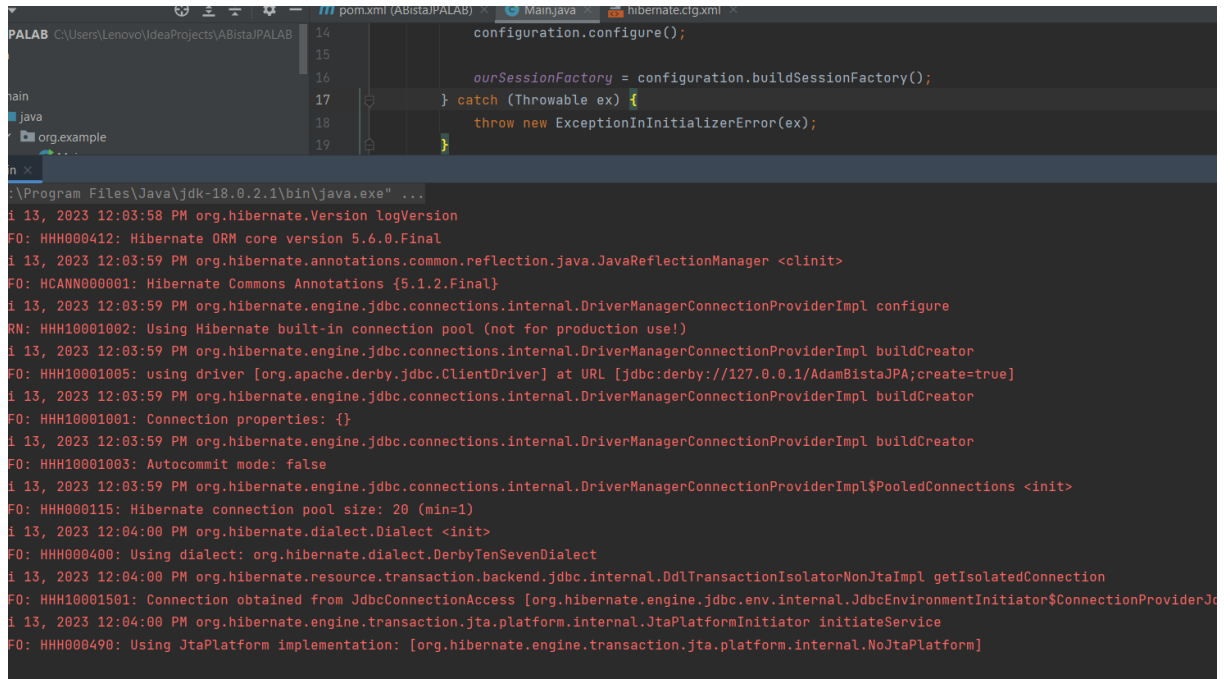
    static {
        try {
            Configuration configuration = new
Configuration();
            configuration.configure();

            ourSessionFactory =
configuration.buildSessionFactory();
        } catch (Throwable ex) {
            throw new ExceptionInInitializerError(ex);
        }
    }

    public static Session getSession() throws
HibernateException {
        return ourSessionFactory.openSession();
    }

    public static void main(final String[] args) throws
Exception {
        final Session session = getSession();
        try {
        } finally {
            session.close();
        }
    }
}
```

12. Błąd przy próbie kompilacji:



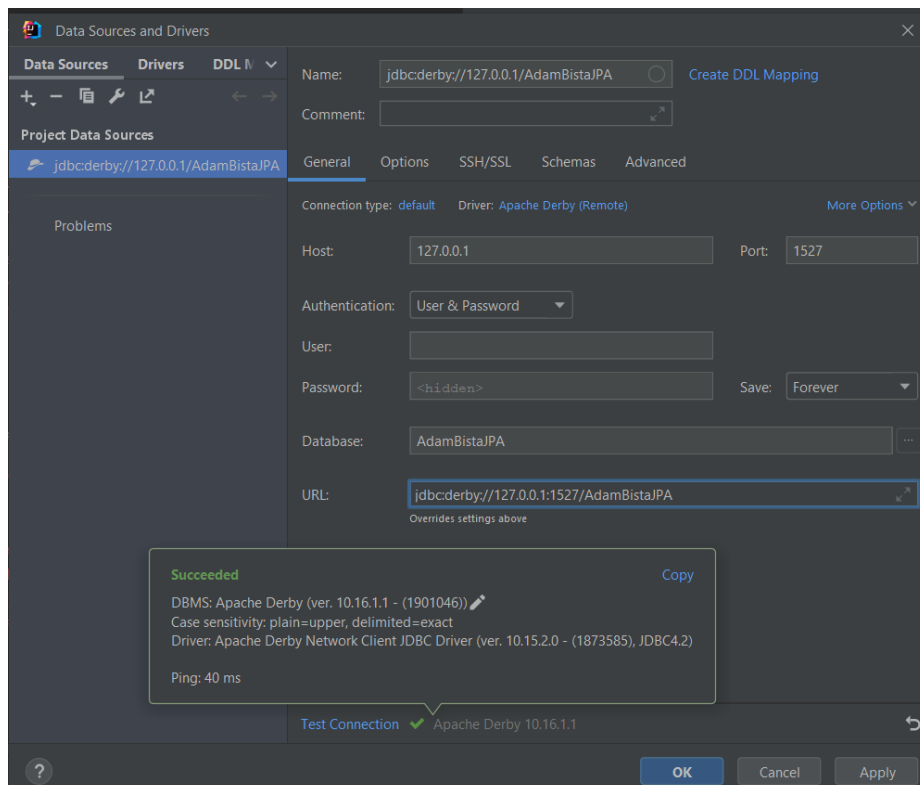
The screenshot shows an IDE with a Java file named `Main.java` and a console window. The Java file contains the following code:

```
14 configuration.configure();
15
16 ourSessionFactory = configuration.buildSessionFactory();
17 } catch (Throwable ex) {
18     throw new ExceptionInInitializerError(ex);
19 }
```

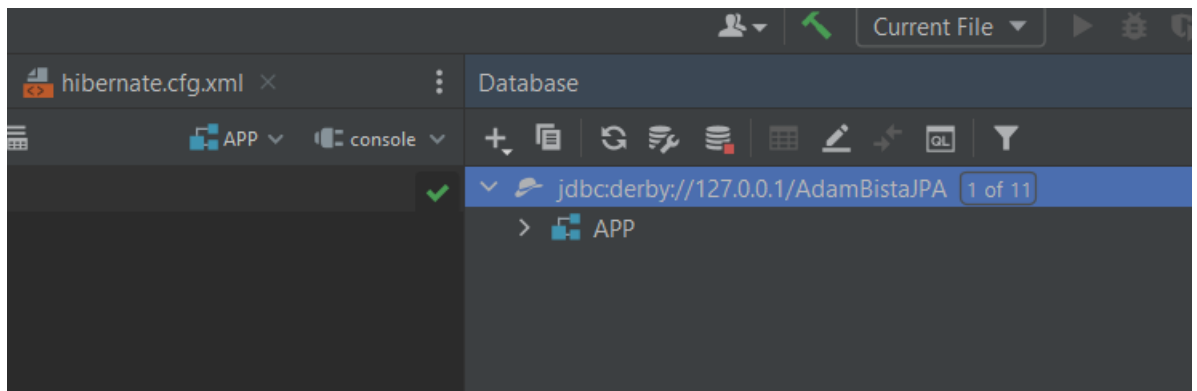
The console window displays the following logs:

```
i 13, 2023 12:03:58 PM org.hibernate.Version logVersion
F0: HHH0000412: Hibernate ORM core version 5.6.0.Final
i 13, 2023 12:03:59 PM org.hibernate.annotations.common.reflection.java.JavaReflectionManager <clinit>
F0: HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
i 13, 2023 12:03:59 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl configure
RN: HHH10001002: Using Hibernate built-in connection pool (not for production use!)
i 13, 2023 12:03:59 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
F0: HHH10001005: using driver [org.apache.derby.jdbc.ClientDriver] at URL [jdbc:derby://127.0.0.1/AdamBistaJPA;create=true]
i 13, 2023 12:03:59 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
F0: HHH10001001: Connection properties: {}
i 13, 2023 12:03:59 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
F0: HHH10001003: Autocommit mode: false
i 13, 2023 12:03:59 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledConnections <init>
F0: HHH000115: Hibernate connection pool size: 20 (min=1)
i 13, 2023 12:04:00 PM org.hibernate.dialect.Dialect <init>
F0: HHH000400: Using dialect: org.hibernate.dialect.DerbyTenSevenDialect
i 13, 2023 12:04:00 PM org.hibernate.resource.transaction.backend.jdbc.internal.OdlTransactionIsolatorNonJtaImpl getIsolatedConnection
F0: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$ConnectionProviderJc
i 13, 2023 12:04:00 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
F0: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
```

13. Ustalamy połączenie



14. Widoczny schemat app:



15. Praca z modelem

Klasa product:

```
package org.example;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity

public class Product {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private int productID;

    public Product(String productName, int unitsOnStock) {
        ProductName = productName;
        UnitsOnStock = unitsOnStock;
    }
    public Product() {

    }

    public int getUnitsOnStock() {
        return UnitsOnStock;
    }

    private String ProductName;
    private int UnitsOnStock;

    public void setUnitsOnStock(int unitsOnStock) {
        UnitsOnStock = unitsOnStock;
    }
}
```

```

@Override
public String toString() {
    return "Product{" +
        "ProductName='" + ProductName + '\'' +
        ", UnitsOnStock=" + UnitsOnStock +
        '}';
}
}

```

16. Main:

```

17. package org.example;

import org.hibernate.HibernateException;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;

public class Main {

    private static final SessionFactory
    ourSessionFactory;

    static {
        try {
            Configuration configuration = new
            Configuration();
            configuration.configure();

            ourSessionFactory =
            configuration.buildSessionFactory();
        } catch (Throwable ex) {
            throw new ExceptionInInitializerError(ex);
        }
    }

    public static Session getSession() throws
    HibernateException {
        return ourSessionFactory.openSession();
    }

    public static void main(final String[] args) throws
    Exception {
        final Session session = getSession();

        Product product = new Product("Krzeslo", 111);
        try {
            Transaction tx = session.beginTransaction();
            session.save(product);
            tx.commit();

```

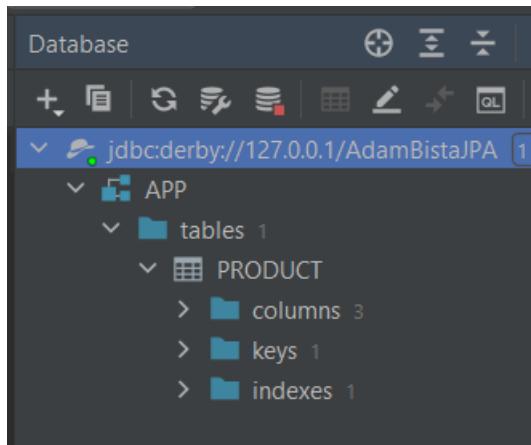


```

    } finally {
        session.close();
    }
}
}

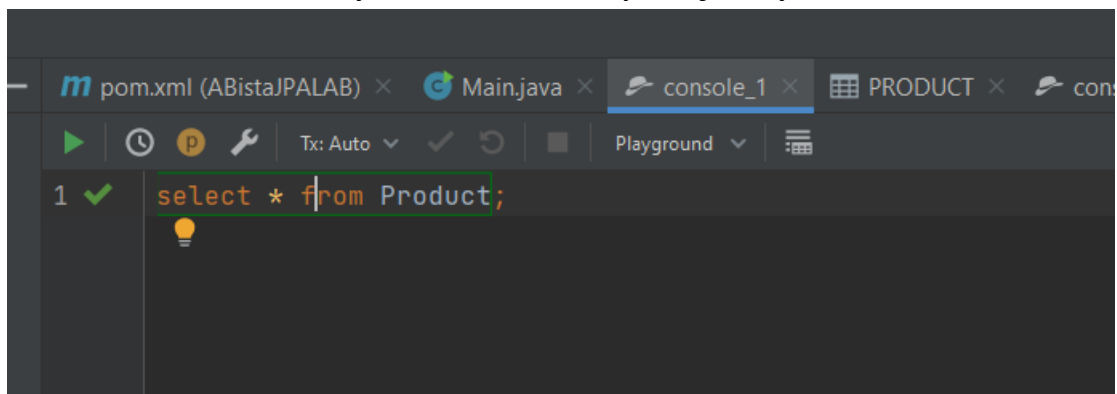
```

18. Nasza tabela w bazie wygląda następująco:



Trzykrotnie wywołujemy main'a

19. Uruchamiamy konsolkę i wpisujemy:



20. Wyniki powyższego wywołania:

	PRODUCTID	PRODUCTNAME	UNITSONSTOCK
1	1	Krzesło	111
2	2	Krzesło	111
3	3	Krzesło	111

	PRODUCTID	PRODUCTNAME	UNITSONSTOCK
1	1	Krzesło	111
2	2	Krzesło	111
3	3	Krzesło	111