Java Hibernate – zadanie domowe

grupa czwartek B 14;40 - 16;15

Dawid Białka 2019/2020

Podczas laboratorium zrobiłem zadania do V punktu włącznie.

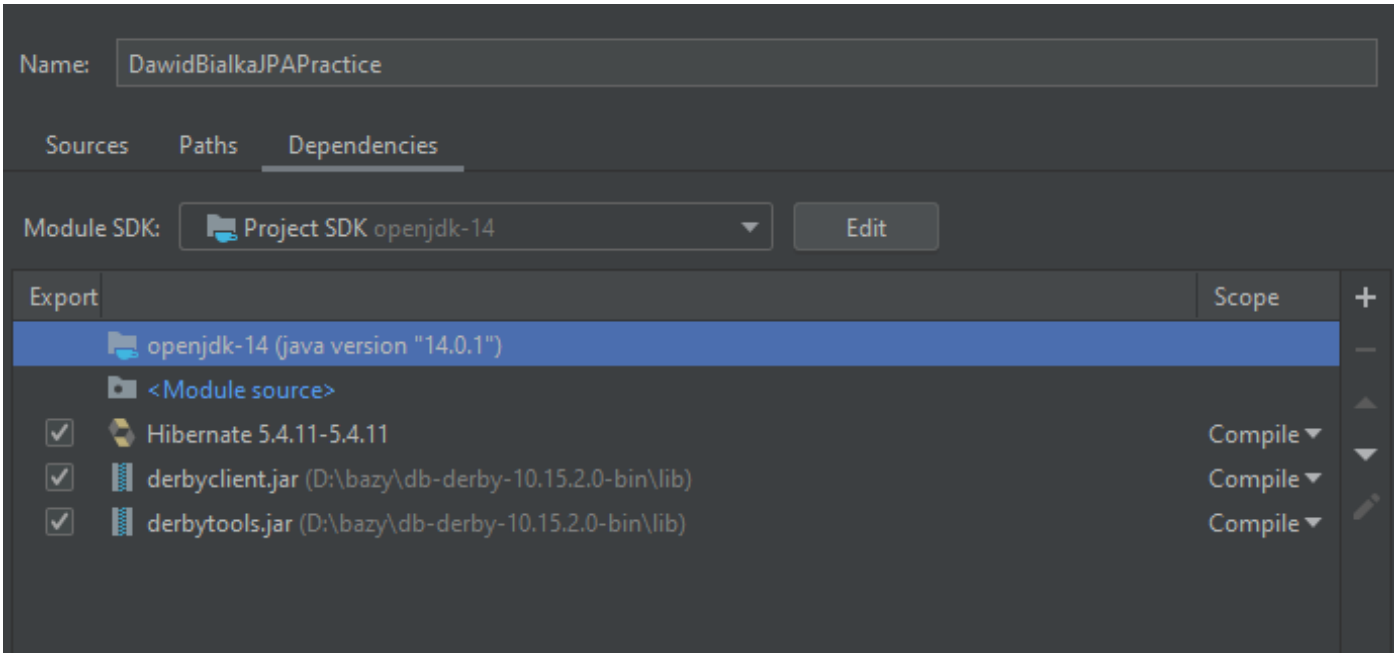
II.

C:\WINDOWS\System32\cmd.exe

```
Thu May 07 12:50:07 CEST 2020 : Security manager installed using the Basic server security policy.
Thu May 07 12:50:11 CEST 2020 : Serwer sieciowy Apache Derby - 10.15.2.0 - (1873585) uruchomiony i gotowy do zaakceptowa
nia po|cze˜ na porcie 1527 w {3}
```

C:\WINDOWS\System32\cmd.exe

```
ij> connect 'jdbc:derby://127.0.0.1/BialkaDawidJPA';
ij> show tables;
TABLE_SCHEM         |TABLE_NAME                    |REMARKS
------------------------------------------------------------------------
SYS                 |SYSALIASES                    |
SYS                 |SYSCHECKS                     |
SYS                 |SYSCOLPERMS                   |
SYS                 |SYSCOLUMNS                    |
SYS                 |SYSCONGLOMERATES              |
SYS                 |SYSCONSTRAINTS                |
SYS                 |SYSDEPENDS                    |
SYS                 |SYSFILES                      |
SYS                 |SYSFOREIGNKEYS                |
SYS                 |SYSKEYS                       |
SYS                 |SYSPERMS                      |
SYS                 |SYSROLES                      |
SYS                 |SYSROUTINEPERMS               |
SYS                 |SYSSCHEMAS                    |
SYS                 |SYSSEQUENCES                  |
SYS                 |SYSSTATEMENTS                 |
SYS                 |SYSSTATISTICS                 |
SYS                 |SYSTABLEPERMS                 |
SYS                 |SYSTABLES                     |
SYS                 |SYSTRIGGERS                   |
SYS                 |SYSUSERS                      |
SYS                 |SYSVIEWS                      |
SYSIBM              |SYSDUMMY1                     |

23 wierszy wybranych
```




```java
import javax.persistence.*;

@Entity
public class Product {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name="ID")
    private  int dbID;

    @Column(name="PRODUCT_NAME")
    private String ProductName;
    @Column(name="UNITS_ON_STOCK")
    private int UnitsOnStock;

    @ManyToOne
    @JoinColumn(name="SUPPLIED_BY")
    private Supplier suppliedBy;

    public Product() {}

    public Product(String name, int units) {
        this.ProductName = name;
        this.UnitsOnStock = units;
    }
}
```

ManyToOne, JoinColumn i private Supplier do następnego punktu.

III.

Main.java | hibernate.cfg.xml | Product.java

```xml
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
        "-//Hibernate/Hibernate Configuration DTD//EN"
        "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
    <session-factory>
        <property name="connection.url">jdbc:derby://127.0.0.1/BialkaDawidJPA</property>
        <property name="connection.driver_class">org.apache.derby.jdbc.ClientDriver</property>
        <property name="dialect">org.hibernate.dialect.DerbyTenSevenDialect</property>
        <property name="format_sql">true</property>
        <property name="show_sql">true</property>
        <property name="use_sql_comments">true</property>
        <!-- DB schema will be updated if needed -->
        <property name="hibernate.hbm2ddl.auto">update</property>
        <mapping class="Product"></mapping>
    </session-factory>
</hibernate-configuration>
```

```java
    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 {
            Product product = new Product( name: "Kawa",  units: 5);
            Transaction tx = session.beginTransaction();
            session.save(product);
            tx.commit();




            /*System.out.println("querying all the managed entities...");
            final Metamodel metamodel = session.getSessionFactory().getMetamodel();
            for (EntityType<?> entityType : metamodel.getEntities()) {
                final String entityName = entityType.getName();
                final Query query = session.createQuery("from " + entityName);
                System.out.println("executing: " + query.getQueryString());
                for (Object o : query.list()) {
                    System.out.println("  " + o);
                }
            }*/
        } finally {
            session.close();
```

```
select * from PRODUCT
```

Output | APP.PRODUCT

1 row | Tx: Auto | DDL

	ID	PRODUCT_NAME	UNITS_ON_STOCK	SUPPLIED_BY
1	1	Kawa	5	<null>

IV.

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

    public static void main(final String[] args) throws Exception {
        final Session session = getSession();
        try {
            Product product = new Product( name: "Kawa",  units: 5);
            Supplier supplier = new Supplier( name: "Firma",  city: "Krakow",  street: "Aleje");
            Transaction tx = session.beginTransaction();
            session.save(supplier);
            tx.commit();
```

```java
public static void main(final String[] args) throws Exception {
    final Session session = getSession();
    try {
        //Product product = new Product("Kawa", 5);
        //Supplier supplier = new Supplier("Firma", "Krakow", "Aleje");
        Transaction tx = session.beginTransaction();
        Product coffe= session.get(Product.class, serializable: 1);
        Supplier firma= session.get(Supplier.class, serializable: 2);
        coffe.setSupplier(firma);
        tx.commit();
```

```java
@Entity
public class Supplier {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name="ID")
    private  int dbID;

    @Column(name="COMPANY_NAME")
    private String CompanyName;

    @Column(name="CITY")
    private String City;

    @Column(name="STREET")
    private String Street;



    public Supplier() {}

    public Supplier(String name, String city, String street) {
        this.CompanyName = name;
        this.City = city;
        this.Street = street;
    }

}
```

```java
@Entity
public class Product {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name="ID")
    private  int dbID;

    @Column(name="PRODUCT_NAME")
    private String ProductName;
    @Column(name="UNITS_ON_STOCK")
    private int UnitsOnStock;


    @JoinColumn(name="SUPPLIED_BY")
    private Supplier suppliedBy;

    public Product() {}

    public Product(String name, int units) {
        this.ProductName = name;
        this.UnitsOnStock = units;
    }

    public void setSupplier(Supplier supplier) {
        this.suppliedBy = supplier;
    }
}
```

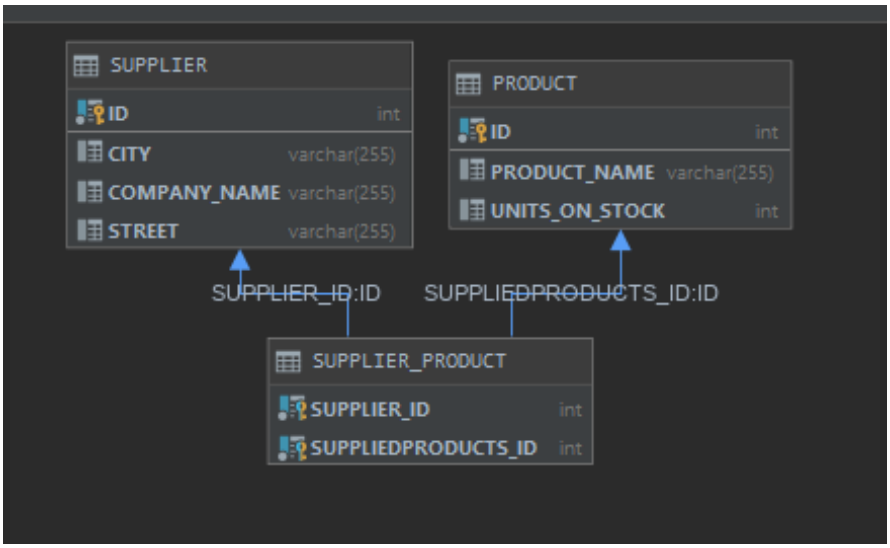
```sql
select * from PRODUCT
join SUPPLIER S on PRODUCT.SUPPLIED_BY = S.ID
```

Output | Result 9

1 row

	ID	PRODUCT_NAME	UNITS_ON_STOCK	SUPPLIED_BY	ID	CITY	COMPANY_NAME	STREET
1	1	Kawa	5	2	2	Krakow	Firma	Aleje

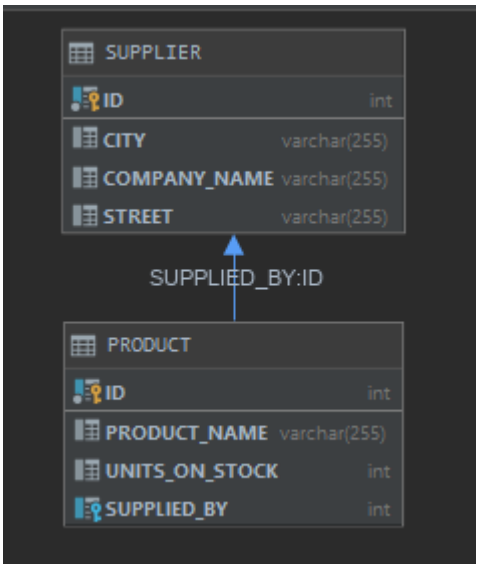
V.



Main.java | Supplier.java | hibernate.cfg.xml | Product.java

```java
import java.util.Set;

@Entity
public class Supplier {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name="ID")
    private  int dbID;

    @Column(name="COMPANY_NAME")
    private String CompanyName;

    @Column(name="CITY")
    private String City;

    @Column(name="STREET")
    private String Street;

    @OneToMany
    private Set<Product> suppliedProducts = new HashSet<>();

    public Supplier() {}

    public Supplier(String name, String city, String street) {
        this.CompanyName = name;
        this.City = city;
        this.Street = street;
    }
}
```

Druga wersja, bez tabeli łącznikowej




```java
@Entity
public class Supplier {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name="ID")
    private  int dbID;

    @Column(name="COMPANY_NAME")
    private String CompanyName;

    @Column(name="CITY")
    private String City;

    @Column(name="STREET")
    private String Street;

    @OneToMany()
    @JoinColumn(name = "SUPPLIED_BY")
    private Set<Product> suppliedProducts = new HashSet<>();

    public Supplier() {}

    public Supplier(String name, String city, String street) {
        this.CompanyName = name;
        this.City = city;
        this.Street = street;
    }
}
```

```java
    }

    public static void main(final String[] args) throws Exception {
        final Session session = getSession();
        try {
            Product product = new Product( name: "Kawa",  units: 5);
            Supplier supplier = new Supplier( name: "Firma",  city: "Krakow",  street: "Aleje");
            Product herbata = new Product( name: "Herbata",  units: 5);
            Transaction tx = session.beginTransaction();
            //Product coffe= session.get(Product.class,1);

            //Supplier firma= session.get(Supplier.class,2);
            //firma.getProducts().add(coffe);
            session.save(herbata);
            session.save(product);
            session.save(supplier);
            tx.commit();
```

```java
    public static void main(final String[] args) throws Exception {
        final Session session = getSession();
        try {
            //Product product = new Product("Kawa", 5);
            //Supplier supplier = new Supplier("Firma", "Krakow", "Aleje");
            //Product product = new Product("Herbata", 5);
            Transaction tx = session.beginTransaction();
            Product coffe = session.get(Product.class, serializable: 9);
            Product herbata = session.get(Product.class,  serializable: 11);

            Supplier firma= session.get(Supplier.class, serializable: 10);
            firma.getProducts().add(coffe);
            firma.getProducts().add(herbata);
            tx.commit();
```

```java
select * from PRODUCT
join SUPPLIER S on PRODUCT.SUPPLIED_BY = S.ID
```

Output | Result 12

2 rows

	ID	PRODUCT_NAME	UNITS_ON_STOCK	SUPPLIED_BY	ID	CITY	COMPANY_NAME	STREET
1	9	Kawa	5	10	10	Krakow	Firma	Aleje
2	11	Herbata	5	10	10	Krakow	Firma	Aleje

VI.

```
    }

    public static void main(final String[] args) throws Exception {
        final Session session = getSession();
        try {
            Product kawa= new Product( name: "Kawa",  units: 5);
            Supplier firma = new Supplier( name: "Firma",  city: "Krakow",  street: "Aleje");
            Product herbata = new Product( name: "Herbata",  units: 5);

            Transaction tx = session.beginTransaction();
            session.save(firma);
            session.save(kawa);
            session.save(herbata);
            tx.commit();

            tx = session.beginTransaction();
            firma.addProduct(kawa);
            herbata.setSupplier(firma);
            tx.commit();
```

```java
@Entity
public class Supplier {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name="ID")
    private  int dbID;

    @Column(name="COMPANY_NAME")
    private String CompanyName;

    @Column(name="CITY")
    private String City;

    @Column(name="STREET")
    private String Street;

    @OneToMany
    @JoinColumn(name = "SUPPLIED_BY")
    private Set<Product> suppliedProducts = new HashSet<>();

    public Supplier() {}

    public Supplier(String name, String city, String street) {
        this.CompanyName = name;
        this.City = city;
        this.Street = street;
    }

    public void addProduct(Product product) {
        this.suppliedProducts.add(product);
        product.setSupplier(this);

    }

}
```

```java
@Entity
public class Product {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name="ID")
    private  int dbID;

    @Column(name="PRODUCT_NAME")
    private String ProductName;
    @Column(name="UNITS_ON_STOCK")
    private int UnitsOnStock;

    public Product() {}

    public Product(String name, int units) {
        this.ProductName = name;
        this.UnitsOnStock = units;
    }

    @ManyToOne
    @JoinColumn(name = "SUPPLIED_BY")
    private Supplier supplier;


    public void setSupplier(Supplier supplier) {
        this.supplier = supplier;
        supplier.addProduct(this);
    }



}
```

```
select * from PRODUCT
join SUPPLIER S on PRODUCT.SUPPLIED_BY = S.ID
```

	ID	PRODUCT_NAME	UNITS_ON_STOCK	SUPPLIED_BY	ID	CITY	COMPANY_NAME	STREET
1	119	Kawa	5	118	118	Krakow	Firma	Aleje
2	120	Herbata	5	118	118	Krakow	Firma	Aleje

VII.

```java
@Entity
public class Category {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name="CategoryID")
    private  int dbID;

    @Column(name="CATEGORY_NAME")
    private String CategoryName;

    @OneToMany
    @JoinColumn(name = "CATEGORY_ID")
    private List<Product> products = new ArrayList<>();

    public Category() {}

    public Category(String name) {
        this.CategoryName = name;
    }

    public void addProduct(Product product) {
        this.products.add(product);
        product.setCategory(this);

    }

}
```

```java
@Entity
public class Product {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name="ID")
    private  int dbID;

    @Column(name="PRODUCT_NAME")
    private String ProductName;
    @Column(name="UNITS_ON_STOCK")
    private int UnitsOnStock;

    public Product() {}

    public Product(String name, int units) {
        this.ProductName = name;
        this.UnitsOnStock = units;
    }

    @ManyToOne
    @JoinColumn(name = "SUPPLIED_BY")
    private Supplier supplier;

    @ManyToOne
    @JoinColumn(name = "CATEGORY_ID")
    private Category category;

    public void setSupplier(Supplier supplier) {
        this.supplier = supplier;
        supplier.addProduct(this);
    }

    public void setCategory(Category category) {
        this.category = category;
        category.addProduct(this);
    }
}
```

```java
public static void main(final String[] args) throws Exception {
    final Session session = getSession();
    try {
        //Product kawa= new Product("Kawa", 5);
        //Supplier firma = new Supplier("Firma", "Krakow", "Aleje");
        //Product herbata = new Product("Herbata", 5);
        Category luksusowe = new Category( name: "Luksusowe");
        Category samochody = new Category( name: "Samochod");
        Product auto1 = new Product( name: "Multipla",  units: 2000);
        Product auto2 = new Product( name: "Lada",  units: 3000);

        Transaction tx = session.beginTransaction();
        session.save(luksusowe);
        session.save(samochody);
        session.save(auto1);
        session.save(auto2);
        Product kawa = session.get(Product.class,  serializable: 119);
        Product herbata = session.get(Product.class,  serializable: 120);
        tx.commit();

        tx = session.beginTransaction();
        auto1.setCategory(samochody);
        auto2.setCategory(sam
        kawa.setCategory(luksusowe);
        luksusowe.addProduct(herbata);

        tx.commit();
```

Product product

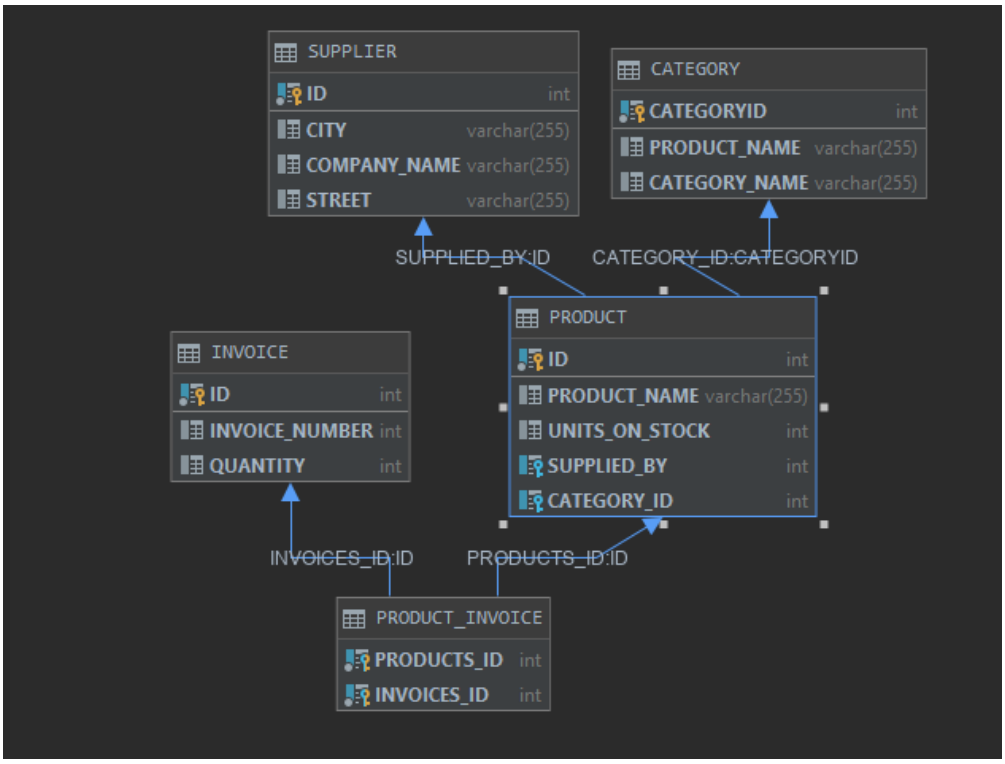
```sql
select * from PRODUCT
join CATEGORY C on PRODUCT.CATEGORY_ID = C.CATEGORYID
```

Output | Result 2

4 rows

	ID	PRODUCT_NAME	UNITS_ON_STOCK	SUPPLIED_BY	CATEGORY_ID	CATEGORYID	PRODUCT_NAME
1	119	Kawa	5	118	121	121	Luksusowe
2	120	Herbata	5	118	121	121	Luksusowe
3	123	Multipla	2000	<null>	122	122	Samochod
4	124	Lada	3000	<null>	122	122	Samochod

VIII.




```java
@Entity
public class Invoice {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name="ID")
    private  int dbID;

    @Column(name="INVOICE_NUMBER")
    private int invoiceNumber;

    @Column(name="QUANTITY")
    private int quantity;

    @ManyToMany(mappedBy = "invoices")
    private Set<Product> products = new HashSet<>();

    public Invoice() {}

    public Invoice(int number, int quantity) {
        this.invoiceNumber = number;
        this.quantity = quantity;
    }

    public void addProduct(Product product) {
        products.add(product);
        product.addInvoice(this);
    }

}
```

```java
    @Column(name="PRODUCT_NAME")
    private String ProductName;
    @Column(name="UNITS_ON_STOCK")
    private int UnitsOnStock;

    public Product() {}

    public Product(String name, int units) {
        this.ProductName = name;
        this.UnitsOnStock = units;
    }

    @ManyToOne
    @JoinColumn(name = "SUPPLIED_BY")
    private Supplier supplier;

    @ManyToOne
    @JoinColumn(name = "CATEGORY_ID")
    private Category category;

    @ManyToMany
    @JoinColumn(name = "INVOICE_ID")
    private Set<Invoice> invoices = new HashSet<>();

    public void setSupplier(Supplier supplier) {
        this.supplier = supplier;
        supplier.addProduct(this);
    }

    public void setCategory(Category category) {
        this.category = category;
        category.addProduct(this);
    }

    public void addInvoice(Invoice invoice) {
        invoices.add(invoice);

    }
}
```

```java
public static void main(final String[] args) throws Exception {
    final Session session = getSession();
    try {
        /*Product kawa= new Product("Kawa", 5);
        Supplier firma = new Supplier("Firma", "Krakow", "Aleje");
        Product herbata = new Product("Herbata", 5);
        Category luksusowe = new Category("Luksusowe");
        Category samochody = new Category("Samochod");
        Product auto1 = new Product("Multipla", 2000);
        Product auto2 = new Product("Lada", 3000);*/
        Invoice invoice1 = new Invoice( number: 1,  quantity: 5);
        Invoice invoice2 = new Invoice( number: 2,  quantity: 7);

        Transaction tx = session.beginTransaction();
        session.save(invoice1);
        session.save(invoice2);
        tx.commit();

        tx = session.beginTransaction();
        Product kawa = session.get(Product.class,  serializable: 119);
        Product lada = session.get(Product.class,  serializable: 123);
        invoice1.addProduct(kawa);
        kawa.addInvoice(invoice2);
        invoice2.addProduct(lada);
        lada.addInvoice(invoice2);
        tx.commit();
```

Produkty sprzedawane na jednej aukcji:

```sql
select PRODUCT_NAME, INVOICE_NUMBER from PRODUCT
join PRODUCT_INVOICE PI on PRODUCT.ID = PI.PRODUCTS_ID
join INVOICE I on PI.INVOICES_ID = I.ID
    where INVOICE_NUMBER = 1;
```

3 rows

	PRODUCT_NAME	INVOICE_NUMBER
1	Kawa	1
2	Lada	1
3	Multipla	1

Multipla sprzedawana na dwóch aukcjach:

```
select PRODUCT_NAME, INVOICE_NUMBER from PRODUCT
join PRODUCT_INVOICE PI on PRODUCT.ID = PI.PRODUCTS_ID
join INVOICE I on PI.INVOICES_ID = I.ID
    where PRODUCT_NAME = 'Multipla';
```

Output | Result 13

2 rows

	PRODUCT_NAME	INVOICE_NUMBER
1	Multipla	1
2	Multipla	2

X.

DawidBialkaJPAPractice C:\Users\Dawid\IdeaProjects\D
- .idea
- lib
- out
- src
  - META-INF
    - persistence.xml
  - Category
  - hibernate.cfg.xml
  - Invoice
  - Main
  - Main2
  - Product
  - Supplier
- DawidBialkaJPAPractice.iml
- External Libraries
- Scratches and Consoles

```xml
<?xml version="1.0"?>
<persistence xmlns="http://java.sun.com/xml/ns/persistence"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd"
             version="2.0">
    <persistence-unit name="myDatabaseConfig"
                      transaction-type="RESOURCE_LOCAL">
        <properties>
            <property name="hibernate.connection.driver_class"
                      value="org.apache.derby.jdbc.ClientDriver"/>
            <property name="hibernate.connection.url"
                      value="jdbc:derby://127.0.0.1/BialkaDawidJPA3"/>
            <property name="hibernate.show_sql" value="true" />
            <property name="hibernate.format_sql" value="true" />
            <property name="hibernate.hbm2ddl.auto" value="create" />
        </properties>
    </persistence-unit>
</persistence>
```

```java
import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.EntityTransaction;
import javax.persistence.Persistence;

public class Main2 {
    public static void main(String[] args) {
        EntityManagerFactory emf = Persistence.
                createEntityManagerFactory( persistenceUnitName: "myDatabaseConfig");
        EntityManager em = emf.createEntityManager();
        EntityTransaction etx = em.getTransaction();
        etx.begin();

        Product auto1 = new Product( name: "Mustang", units: 50);
        Product auto2 = new Product( name: "Syrenka",  units: 20);
        Supplier firma = new Supplier( name: "Firma samochodowa",  city: "Warszawa",  street: "Ulica");

        em.persist(auto1);
        em.persist(auto2);
        em.persist(firma);

        firma.addProduct(auto1);
        auto2.setSupplier(firma);

        etx.commit();


    }
}
```

```
SELECT * FROM PRODUCT
```

Output | APP.PRODUCT

2 rows | Tx: Auto | DDL

	ID	PRODUCT_NAME	UNITS_ON_STOCK	CATEGORY_ID	SUPPLIED_BY
1	1	Mustang	50	<null>	3
2	2	Syrenka	20	<null>	3

Product i Supplier bez zmian.

XI.

```java
    public Product(String name, int units) {
        this.ProductName = name;
        this.UnitsOnStock = units;
    }

    @ManyToOne
    @JoinColumn(name = "SUPPLIED_BY")
    private Supplier supplier;

    @ManyToOne
    @JoinColumn(name = "CATEGORY_ID")
    private Category category;

    @ManyToMany(cascade = CascadeType.PERSIST)
    private Set<Invoice> invoices = new HashSet<>();

    public void setSupplier(Supplier supplier) {
        this.supplier = supplier;
        supplier.addProduct(this);
    }

    public void setCategory(Category category) {
        this.category = category;
        category.addProduct(this);
    }

    public void addInvoice(Invoice invoice) {
        invoices.add(invoice);

    }
}
```

```java
@Entity
public class Invoice {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name="ID")
    private  int dbID;

    @Column(name="INVOICE_NUMBER")
    private int invoiceNumber;

    @Column(name="QUANTITY")
    private int quantity;

    @ManyToMany(mappedBy = "invoices", cascade = {CascadeType.PERSIST})
    private Set<Product> products = new HashSet<>();

    public Invoice() {}

    public Invoice(int number, int quantity) {
        this.invoiceNumber = number;
        this.quantity = quantity;
    }

    public void addProduct(Product product) {
        products.add(product);
        product.addInvoice(this);
    }

}
```

```java
public class Main2 {
    public static void main(String[] args) {
        EntityManagerFactory emf = Persistence.
                createEntityManagerFactory( persistenceUnitName: "myDatabaseConfig");
        EntityManager em = emf.createEntityManager();
        EntityTransaction etx = em.getTransaction();
        etx.begin();

        Product auto1 = new Product( name: "Mustang", units: 50);
        Product auto2 = new Product( name: "Syrenka",  units: 20);
        Product auto3 = new Product( name: "UAZ",  units: 25);
        Invoice invoice1 = new Invoice( number: 1,  quantity: 5);
        Invoice invoice2 = new Invoice( number: 2,  quantity: 10);
        Invoice invoice3 = new Invoice( number: 3,  quantity: 20);

        auto3.addInvoice(invoice2);
        auto3.addInvoice(invoice3);
        invoice1.addProduct(auto1);
        invoice1.addProduct(auto2);

        em.persist(invoice1);
        em.persist(auto3);

        etx.commit();


    }
```

```java
select PRODUCT_NAME, INVOICE_NUMBER from PRODUCT
join PRODUCT_INVOICE PI on PRODUCT.ID = PI.PRODUCTS_ID
join INVOICE I on PI.INVOICES_ID = I.ID;

SELECT * FROM PRODUCT
```

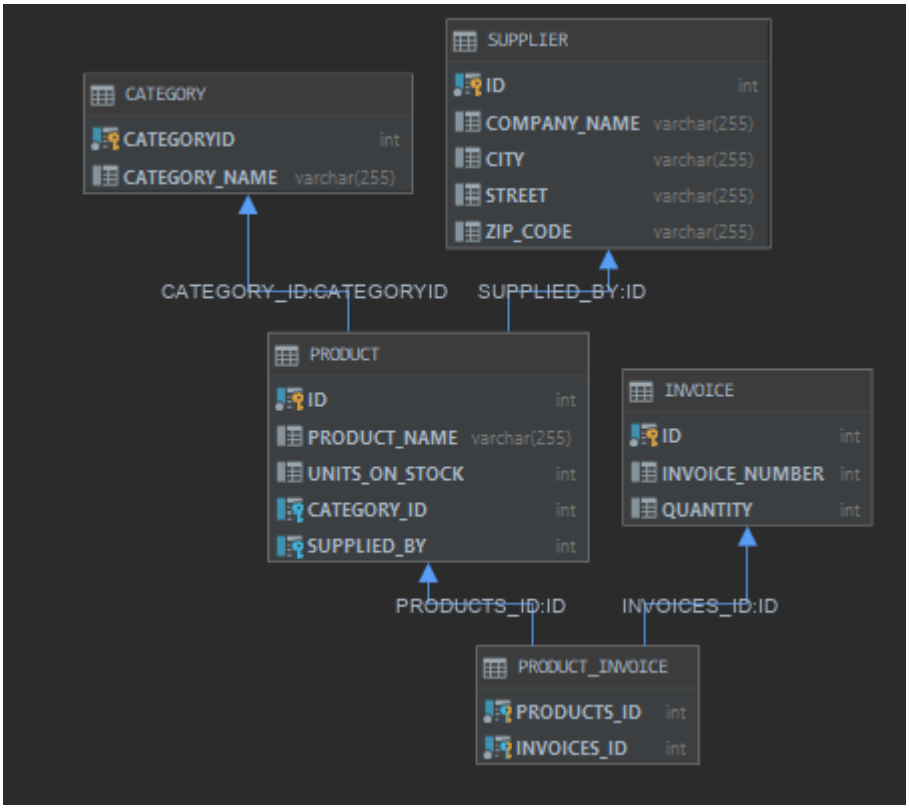
	PRODUCT_NAME	INVOICE_NUMBER
1	Syrenka	1
2	Mustang	1
3	UAZ	2
4	UAZ	3

XII.

Wbudowanie Address do Supplier

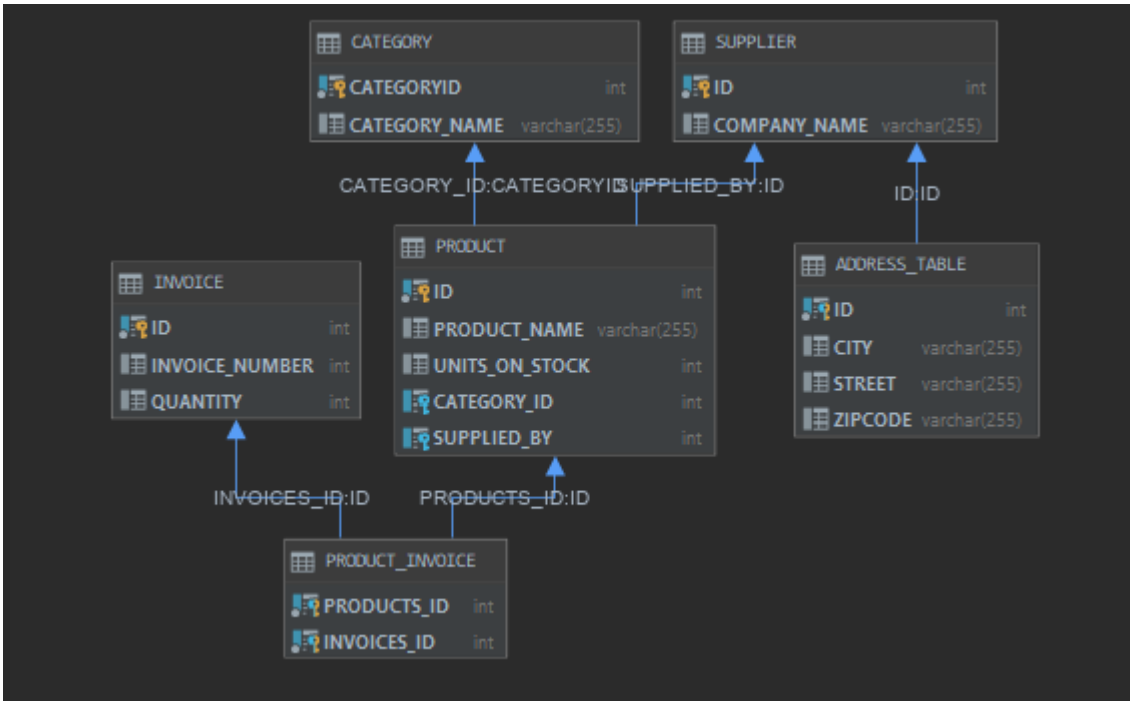
```java
@Embeddable
public class Address {

    @Column(name="CITY")
    private String City;

    @Column(name="STREET")
    private String Street;

    @Column(name="ZIP_CODE")
    private String zipCode;

    public Address() {}

    public Address(String city, String street, String zipCode) {
        this.City = city;
        this.Street = street;
        this.zipCode = zipCode;
    }
}
```

```java
@Entity
public class Supplier {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name="ID")
    private  int dbID;

    @Column(name="COMPANY_NAME")
    private String CompanyName;

    @Embedded
    private Address address;

    @OneToMany
    @JoinColumn(name = "SUPPLIED_BY")
    private Set<Product> suppliedProducts = new HashSet<>();

    public Supplier() {}

    public Supplier(String name, Address address) {
        this.CompanyName = name;
        this.address = address;
    }

    public void addProduct(Product product) {
        this.suppliedProducts.add(product);
        product.setSupplier(this);

    }
```


SUPPLIER
ID int
COMPANY_NAME varchar(255)
CITY varchar(255)
STREET varchar(255)
ZIP_CODE varchar(255)
CATEGORY
CATEGORYID int
CATEGORY_NAME varchar(255)
CATEGORY_ID:CATEGORYID
SUPPLIED_BY:ID
PRODUCT
ID int
PRODUCT_NAME varchar(255)
UNITS_ON_STOCK int
CATEGORY_ID int
SUPPLIED_BY int
INVOICE
ID int
INVOICE_NUMBER int
QUANTITY int
PRODUCTS_ID:ID
INVOICES_ID:ID
PRODUCT_INVOICE
PRODUCTS_ID int
INVOICES_ID int

Zmapowanie klasy do dwóch tabel.

```java
@Entity
@SecondaryTable(name = "ADDRESS_TABLE")
public class Supplier {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name="ID")
    private  int dbID;

    @Column(name="COMPANY_NAME")
    private String CompanyName;

    @Column(table="ADDRESS_TABLE")
    private String City;

    @Column(table="ADDRESS_TABLE")
    private String Street;

    @Column(table="ADDRESS_TABLE")
    private String zipCode;

    @OneToMany
    @JoinColumn(name = "SUPPLIED_BY")
    private Set<Product> suppliedProducts = new HashSet<>();

    public Supplier() {}

    public Supplier(String name, String city, String street, String zipCode) {
        this.City = city;
        this.Street = street;
        this.zipCode = zipCode;
    }

    public void addProduct(Product product) {
        this.suppliedProducts.add(product);
        product.setSupplier(this);

    }

}
```


CATEGORY
CATEGORYID int
CATEGORY_NAME varchar(255)
SUPPLIER
ID int
COMPANY_NAME varchar(255)
CATEGORY_ID:CATEGORYID
SUPPLIED_BY:ID
ID:ID
PRODUCT
ID int
PRODUCT_NAME varchar(255)
UNITS_ON_STOCK int
CATEGORY_ID int
SUPPLIED_BY int
INVOICE
ID int
INVOICE_NUMBER int
QUANTITY int
ADDRESS_TABLE
ID int
CITY varchar(255)
STREET varchar(255)
ZIPCODE varchar(255)
INVOICES_ID:ID
PRODUCTS_ID:ID
PRODUCT_INVOICE
PRODUCTS_ID int
INVOICES_ID int

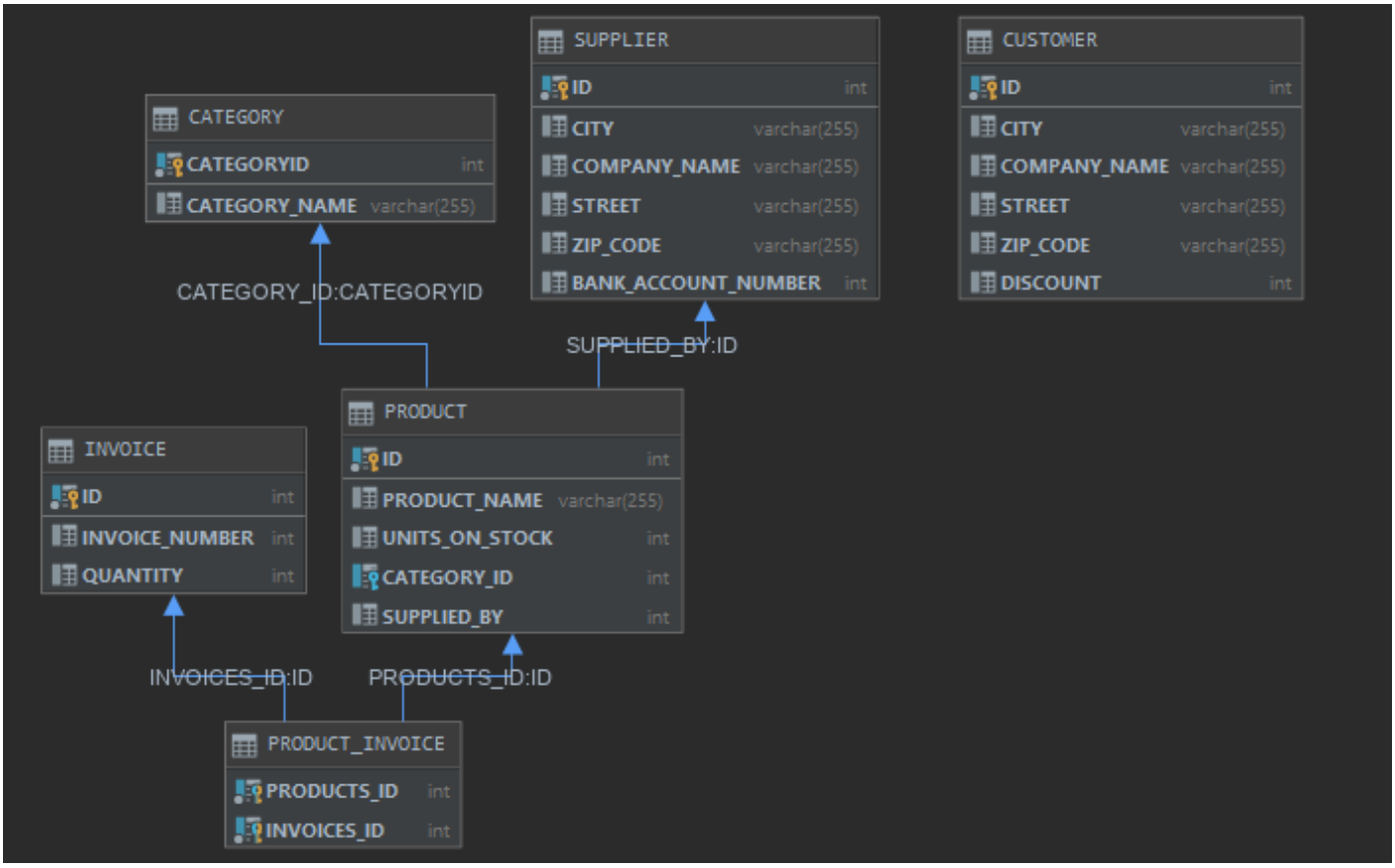
XIII.

Table per Class

```java
@Entity
@Inheritance(strategy= InheritanceType.TABLE_PER_CLASS)
public abstract class Company {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name="ID")
    private  int dbID;

    @Column(name="COMPANY_NAME")
    private String CompanyName;

    @Column(name="CITY")
    private String City;

    @Column(name="STREET")
    private String Street;

    @Column(name="ZIP_CODE")
    private String zipCode;

    public Company() {}

    public Company(String name, String city, String street, String zipCode) {
        this.City = city;
        this.Street = street;
        this.zipCode = zipCode;
    }

}
```

```java
@Entity
public class Customer extends Company {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name="ID")
    private  int dbID;

    @Column(name="DISCOUNT")
    private int discount;

    public Customer() {}

    public Customer(String name, String city, String street, String zipCode, int bankAccountNumber) {
        super(name, city, street, zipCode);
        this.discount = bankAccountNumber;
    }


}
```

```java
@Entity
public class Supplier extends Company{
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name="ID")
    private  int dbID;

    @Column(name="BANK_ACCOUNT_NUMBER")
    private int bankAccountNumber;

    @OneToMany
    @JoinColumn(name = "SUPPLIED_BY")
    private Set<Product> suppliedProducts = new HashSet<>();

    public Supplier() {}

    public Supplier(String name, String city, String street, String zipCode, int bankAccountNumber) {
        super(name, city, street, zipCode);
        this.bankAccountNumber = bankAccountNumber;
    }

    public void addProduct(Product product) {
        this.suppliedProducts.add(product);
        product.setSupplier(this);

    }

}
```



```
SELECT * FROM CUSTOMER;
```

Output | APP.CUSTOMER

1 row | Tx: Auto | DDL

	ID	CITY	COMPANY_NAME	STREET	ZIP_CODE	DISCOUNT
1	2	Jakies miasto	Jakis klient	Jakas ulica	720-32	132

```
SELECT * FROM SUPPLIER;
```

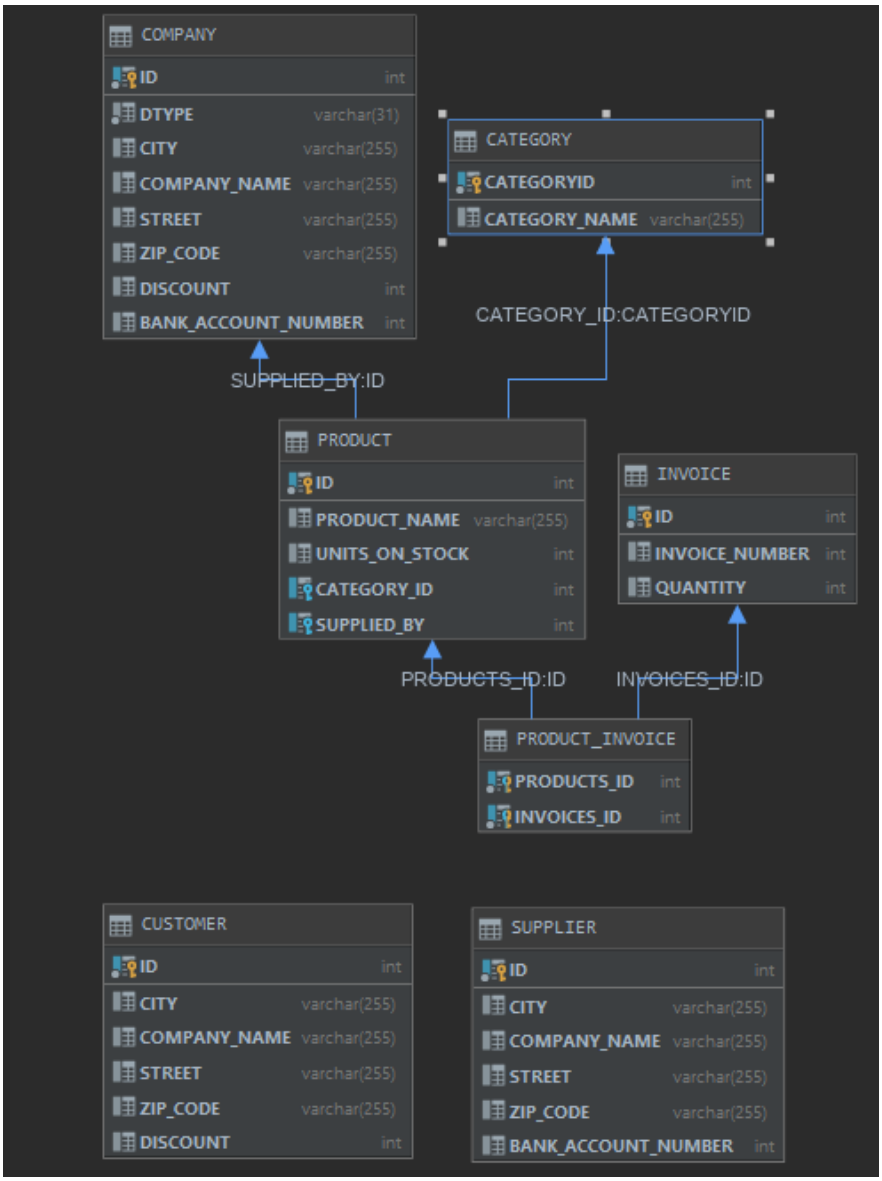
Output | APP.SUPPLIER

1 row | Tx: Auto | DDL

	ID	CITY	COMPANY_NAME	STREET	ZIP_CODE	BANK_ACCOUNT_NUMBER
1	1	Warszawa	Jakas firma	Ulica jakas	32-720	66

Single table

```
@Entity
@Inheritance(strategy= InheritanceType.SINGLE_TABLE)
public abstract class Company {
```



```
SELECT * FROM COMPANY;
```

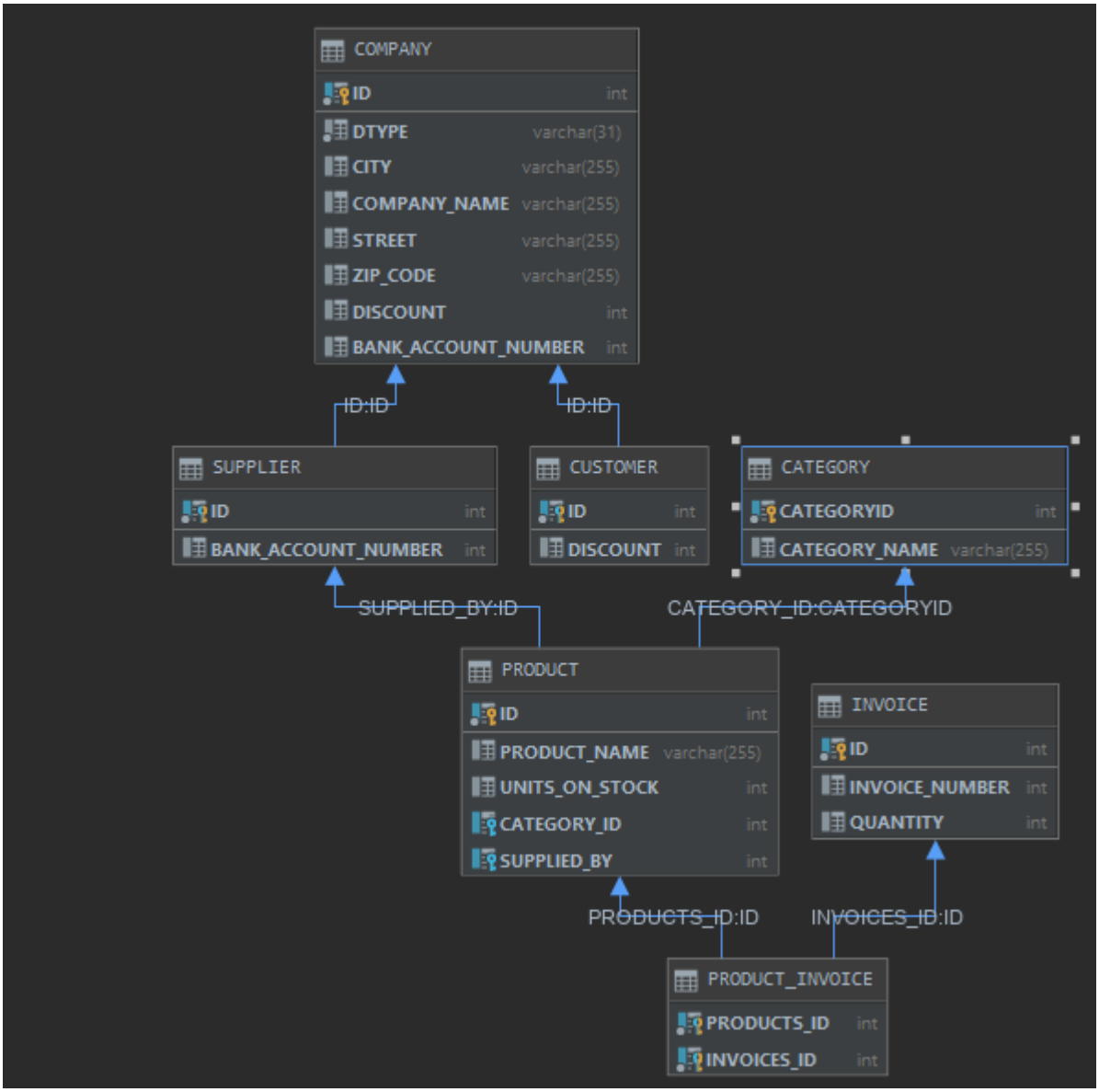
Output | APP.COMPANY

2 rows | Tx: Auto | DDL | Tab-se...d (TSV

	DTYPE	ID	CITY	COMPANY_NAME	STREET	ZIP_CODE	DISCOUNT	BANK_ACCOUNT_NUMBER
1	Supplier	1	Warszawa	Jakas firma	Ulica jakas	32-720	<null>	66
2	Customer	2	Jakies miasto	Jakis klient	Jakas ulica	720-32	132	<null>

Joined

```java
@Entity
@Inheritance(strategy= InheritanceType.JOINED)
public abstract class Company {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name="ID")
```



```
SELECT * FROM COMPANY;
```

Output | APP.COMPANY

2 rows | Tx: Auto | DDL | Tab-se...d (TSV

	DTYPE	ID	CITY	COMPANY_NAME	STREET	ZIP_CODE	DISCOUNT	BANK_ACCOUNT_NUMBER
1	Supplier	1	Warszawa	Jakas firma	Ulica jakas	32-720	<null>	66
2	Customer	2	Jakies miasto	Jakis klient	Jakas ulica	720-32	132	<null>