Hibernate - Piotr Witek

Zadanie 2 - ManyToOne

```java
package com.company;

import javax.persistence.*;

@Entity
public class Product {
    @Id
    @GeneratedValue (strategy = GenerationType.AUTO)
    private int ProductID;
    private String ProductName;

    private Integer UnitsOnStock;

    @ManyToOne
    @JoinColumn(name="SupplierID",nullable = false)
    private Supplier supplier;

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

```java
    public static void main(final String[] args) throws Exception {

        try (Session session = getSession()) {
//          Product product = new Product("Table",10);
//          Supplier supplier = new Supplier("OoohBi","Krakowska","Krakow");
            Transaction tx = session.beginTransaction();
//          session.save(supplier);
//          session.save(product);
            tx.commit();
        }
    }
}
```

	ProductID	ProductName	UnitsOnStock	SupplierID
1	1	Table	10	1
2	2	Chair	15	1
3	3	Drawer	23	1

	SupplierID 1	City	CompanyName	Street
1	1	Krakow	OoohBi	Krakowska

```java
try (Session session = getSession()) {
//          Product product = new Product("Table",10);
//          Supplier supplier = new Supplier("OoohBi","Krakowska","Krakow");
        Transaction tx = session.beginTransaction();
//          session.save(supplier);
//          session.save(product);
        Query query = session.createQuery( s: "From Product");
        List results = query.getResultList();
        for (Object result : results) {
            System.out.println(result);
        }
```

Testowanie select'ów i połączenia ManyToOne

```java
        Query query = session.createQuery( s: " From Product P where P.supplier.SupplierID = 1");
        List results = query.getResultList();
        for (Object result : results) {
            System.out.println(((Product)result).getProductName());
```

```
Hibernate:
    select
        supplier0_.SupplierID as supplier1_1_0_,
        supplier0_.City as city2_1_0_,
        supplier0_.CompanyName as companyn3_1_0_,
        supplier0_.Street as street4_1_0_
    from
        Supplier supplier0_
    where
        supplier0_.SupplierID=?
Table
Chair
Drawer
```

Zadanie 3 - OneToMany

Bez tabeli łącznikowej

```java
@Entity
public class Supplier {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private int SupplierID;
    private String CompanyName;
    private String Street;
    private String City;

    @OneToMany
    private Set<Product> products;

    public Set<Product> getProducts() { return products; }

    public Supplier() {
    }

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

```
32
33          try (Session session = getSession()) {
34              //  Product product = new Product("Table",10);
35              //  Supplier supplier = new Supplier("OoohBi","Krakowska","Krakow");
36              Transaction tx = session.beginTransaction();
37              //  session.save(supplier);
38              //  session.save(product);
39              Query query = session.createQuery( s: " From Supplier");
40              List results = query.getResultList();
41              for (Object result : results) {
42                  Set<Product> productSet = ((Supplier)result).getProducts();
43                  for(Product product:productSet){
44                      System.out.println(product.getProductName());
45                  }
46              }
47          //  tx.commit();
```

```
Run: Main
    where
        products0_.SupplierID=?
Drawer
Chair
Table

Process finished with exit code 0
```

Z tabelą łącznikow

```
@OneToMany(cascade = CascadeType.ALL)
@JoinTable(
        name = "SupplierProducts",
        joinColumns = @JoinColumn(name = "SupplierID"),
        inverseJoinColumns = @JoinColumn(name = "ProductID")
)
private Set<Product> products;
public Set<Product> getProducts() { return products;}
public void setProducts(Set<Product> products) { this.products = products; }
```

```
com.company
    Main
    Product
    Supplier
  hibernate.cfg.xml
.gitignore
HIBJPA_LAB_Prod_2021.pdf
identifier.sqlite
mydb.db
mydb_backup.db
PiotrWitekJPAPractice.iml
```

```
40          List results = query.getResultList();
41          for (Object result : results) {
42              Set<Product> products = ((Supplier)result).getProducts();
43              for(Product product: products){
44                  System.out.println(product.getProductName());
45              }
46              System.out.println("Products of "+((Supplier)result).getCompanyName());
47              System.out.println("****************************");
48
```

Main

```
        product1_.ProductName as productn2_0_1_,
        product1_.UnitsOnStock as unitsons3_0_1_
    from
        SupplierProducts products0_
    inner join
        Product product1_
            on products0_.ProductID=product1_.ProductID
    where
        products0_.SupplierID=?
Drawer
Table
Products of FizzBuzz
****************************
```

	ProductID	ProductName	UnitsOnStock
1	4	Table	10
2	5	Drawer	10
3	6	Chair	23

	SupplierID	City	CompanyName	Street
1	7	Krakow	AGH	Kawiory
2	8	Krakow	FizzBuzz	Krakowska

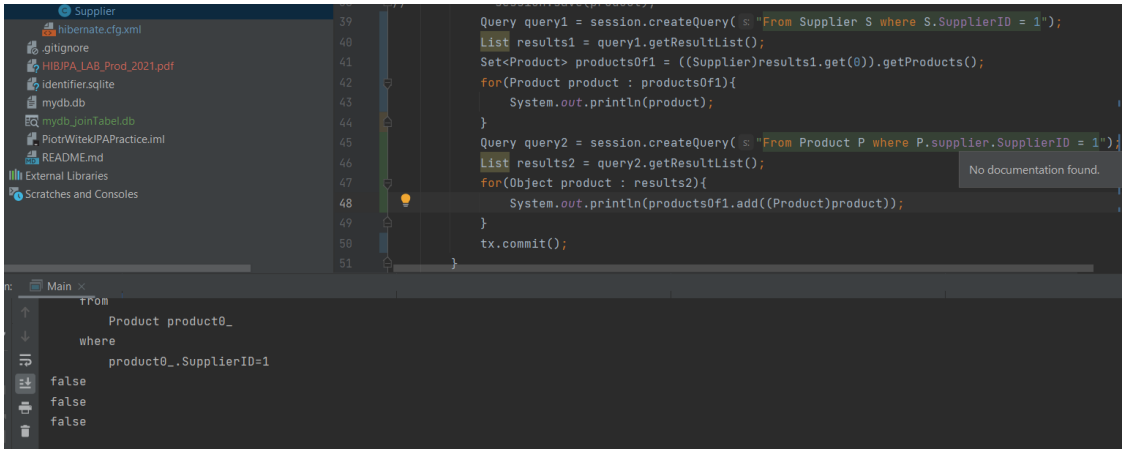
	SupplierID	ProductID
1	8	5
2	8	4
3	7	6

Zadanie 4 ManyToOne & OneToMany

```
@ManyToOne
@JoinColumn(name="SupplierID",nullable = false)
private Supplier supplier;
```

```
@OneToMany(mappedBy="supplier")
private Set<Product> products;
```

Testuję, czy istnieje połączenie z obu stron wykorzystując właściwości Set'u - jeśli spróbujemy dodać te same wartości po raz kolejny metoda Set.add(object) zwróci false



Zadanie 5

Dodaj klase Category z property int CategoryID, String Name oraz listą produktow List Products

```
@Entity
public class Category {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private int CategoryID;
    private String Name;

    @OneToMany
    private List<Product> Products;

    public Category(String name) {
        Name = name;
    }

    public Category() {
    }
```

```java
try (Session session = getSession()) {
    Category category = new Category( name: "AGD");
    Transaction tx = session.beginTransaction();
    session.save(category);
    tx.commit();
}
```

	CategoryID	Name
1	4	Furniture
2	5	RTV
3	6	AGD

```java
    Query query1 = session.createQuery( s: "From Product where ProductID = 3");
    Product product = (Product) query1.getResultList().get(0);
    Query query = session.createQuery( s: "From Category where CategoryID = 5");
    Category category = (Category)query.getResultList().get(0);
    category.getProducts().add(category.getProducts().size(),product);
    System.out.println(category);
    Transaction tx = session.beginTransaction();

      session.save(category);
    tx.commit();
```

	Category_CategoryID	Products_ProductID
1	4	1
2	5	2
3	5	3

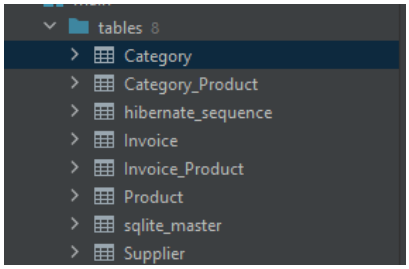
```java
Query query = session.createQuery( s: "From Category where CategoryID = 5");
List categories = query.getResultList();
for(Object category: categories){
    System.out.println(((Category)category).getName());
    for(Product product: ((Category) category).getProducts()){
        System.out.println(product.getProductName());
    }
}
```

```
    where
        products0_.Category_CategoryID=?
Chair
Drawer
```

Zadanie 6 ManyToMany

```java
@ManyToMany
private Set<Product> products;

public Set<Product> getProducts() {
    return products;
}

public void setProducts(Set<Product> products) {
    this.products = products;
}
```

```java
@Entity
public class Invoice {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private int InvoiceID;

    private String InvoiceNumber;
    private int Quantity;

    @ManyToMany
    private Set<Product> products;

    public Set<Product> getProducts() {
        return products;
    }

    public void setProducts(Set<Product> products) {
        this.products = products;
    }

    public Invoice(String invoiceNumber, int quantity) {
        InvoiceNumber = invoiceNumber;
        Quantity = quantity;
    }

    public Invoice() {
    }
```




```java
public void sellProduct(int quantity, Invoice invoice) {
    // nie jestem pewny czy o to chodzi, ale to mało istotne, bo zadanie ma sprawdzać inne rzeczy
    if (quantity > 0 && this.UnitsOnStock >= quantity) {
        UnitsOnStock -= quantity;
        invoice.setQuantity(invoice.getQuantity() + quantity);
        invoice.getProducts().add(this);
    }
}
```

Spróbuje sprzedać produkty i dodać je do faktury

Oto stan przed :

	ProductID	ProductName	UnitsOnStock	SupplierID 1
1	1	Table	10	1
2	2	Chair	15	1
3	3	Drawer	23	1

Stan po:

	ProductID	ProductName	UnitsOnStock	SupplierID 1
1	1	Table	7	1
2	2	Chair	15	1
3	3	Drawer	23	1

	InvoiceID	InvoiceNumber	Quantity
1	1	13231aa	3
2	2	13sd233aa	0

	invoices_InvoiceID	products_ProductID
1	1	1

Po kolejnej sprzedaży

	invoices_InvoiceID	products_ProductID
1	1	1
2	1	2

	InvoiceID	InvoiceNumber	Quantity
1	1	13231aa	11
2	2	13sd233aa	0

	ProductID	ProductName	UnitsOnStock	SupplierID ▲ 1
1	1	Table	7	1
2	2	Chair	7	1
3	3	Drawer	23	1

Pobieranie wszystkich produktów należących do invoice'u:

```java
    try (Session session = getSession()) {
        Query query = session.createQuery( s: "From Invoice");
        List invoices = query.getResultList();
        for (Object invoice: invoices){
            Set<Product> products = ((Invoice) invoice).getProducts();
            for (Product product : products){
                System.out.println(product.getProductName());
            }
            System.out.println(((Invoice)invoice).getInvoiceNumber());
            System.out.println("*******************");
        }
    }
```

Wynik:

```
            on product1_.SupplierID=supplier2_.SupplierID
    where
        products0_.invoices_InvoiceID=?
Table
Chair
13231aa
*******************
Hibernate:
    select
        products0_.invoices_InvoiceID as invoices1_3_0_,
        products0_.products_ProductID as products2_3_0_,
        product1_.ProductID as producti1_4_1_,
        product1_.ProductName as productn2_4_1_,
        product1_.UnitsOnStock as unitsons3_4_1_,
        product1_.SupplierID as supplier4_4_1_,
        supplier2_.SupplierID as supplier1_5_2_,
        supplier2_.City as city2_5_2_,
        supplier2_.CompanyName as companyn3_5_2_,
        supplier2_.Street as street4_5_2_
    from
        Invoice_Product products0_
    inner join
        Product product1_
            on products0_.products_ProductID=product1_.ProductID
    inner join
        Supplier supplier2_
            on product1_.SupplierID=supplier2_.SupplierID
    where
        products0_.invoices_InvoiceID=?
Drawer
Table
13sd233aa
*******************

Process finished with exit code 0
```

Pobieranie wszystkich invoice'ów dla produktu:

```java
public static void main(final String[] args) throws Exception {

    try (Session session = getSession()) {
        Query query = session.createQuery( s: "From Product");
        List products = query.getResultList();
        for (Object product: products){
            Set<Invoice> invoices = ((Product) product).getInvoices();
            for (Invoice invoice : invoices){
                System.out.println(invoice.getInvoiceNumber());
            }
            System.out.println(((Product)product).getProductName());
            System.out.println("*******************");
        }
    }
```

Wynik:

```
    where
        invoices0_.products_ProductID=?
13231aa
Chair
*******************
Hibernate:
    select
        invoices0_.products_ProductID as products2_3_0_,
        invoices0_.invoices_InvoiceID as invoices1_3_0_,
        invoice1_.InvoiceID as invoicei1_2_1_,
        invoice1_.InvoiceNumber as invoicen2_2_1_,
        invoice1_.Quantity as quantity3_2_1_
    from
        Invoice_Product invoices0_
    inner join
        Invoice invoice1_
            on invoices0_.invoices_InvoiceID=invoice1_.InvoiceID
    where
        invoices0_.products_ProductID=?
13sd233aa
Drawer
*******************

Process finished with exit code 0
```

Początek JPA:

Zadanie 7

```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">
        <provider>org.hibernate.ejb.HibernatePersistence</provider>
        <class>org.sample.entities.Entity</class>

        <properties>
            <property name="dialect" value="org.hibernate.dialect.SQLiteDialect" />
            <property name="javax.persistence.jdbc.driver" value="org.sqlite.JDBC" />
            <property name="javax.persistence.jdbc.url" value="jdbc:sqlite:mydb.db" />
            <property name="javax.persistence.jdbc.user" value="" />
            <property name="javax.persistence.jdbc.password" value="" />
            <property name="hibernate.show_sql" value="true" />
            <property name="format_sql" value="true" />
            <property name="hibernate.connection.charSet" value="UTF-8" />
<!--            <property name="hibernate.hbm2ddl.auto" value="create" />-->
        </properties>
    </persistence-unit>
</persistence>
```

```
public class Main {
    public static void main(final String[] args)  {
        EntityManagerFactory emf = Persistence.createEntityManagerFactory( persistenceUnitName: "myDatabaseConfig");
        EntityManager em = emf.createEntityManager();
        EntityTransaction etx = em.getTransaction();
        etx.begin();
        Query query = em.createQuery( s: "FROM Product ");
        List results = query.getResultList();
        for ( Object o : results){
            System.out.println(((Product)o).getProductName());
        }
//        etx.commit();
//        em.close();
```

```
Main
INFO: HHH10001005: using driver [org.sqlite.JDBC] at URL [jdbc:sqlite:mydb.db]
maj 03, 2022 6:11:10 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001001: Connection properties: {charSet=UTF-8, password=****, user=}
maj 03, 2022 6:11:10 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001003: Autocommit mode: false
maj 03, 2022 6:11:10 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledConnections <init>
INFO: HHH000115: Hibernate connection pool size: 20 (min=1)
maj 03, 2022 6:11:11 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
Hibernate: select product0_.ProductID as productid1_4_, product0_.ProductName as productn2_4_, product0_.UnitsOnStock as unitsons3_4_, product0_.SupplierID as supplier4_4_ from Product product0_
Hibernate: select supplier0_.SupplierID as supplier1_5_0_, supplier0_.City as city2_5_0_, supplier0_.CompanyName as companyn3_5_0_, supplier0_.Street as street4_5_0_ from Supplier supplier0_ where supplier0_.SupplierID=?
Table
Chair
Drawer
```

Zadanie 8

Tworzenie nowego produktu przy nowej fakturze.

```
@ManyToMany(cascade = CascadeType.PERSIST)
private Set<Product> products = new HashSet<>();
```

```java
public class Main {
    public static void main(final String[] args)  {
        EntityManagerFactory emf = Persistence.createEntityManagerFactory( persistenceUnitName: "myDatabaseConfig");
        EntityManager em = emf.createEntityManager();
        EntityTransaction etx = em.getTransaction();
        Invoice invoice = new Invoice( invoiceNumber: "test", quantity: 0);
//        Product product = (Product) em.createQuery("FROM Product WHERE ProductID=1").getResultList().get(0);
        Product product = new Product( productName: "testProduct", unitsOnStock: 10);
        product.setSupplier((Supplier) em.createQuery( s: "FROM Supplier WHERE SupplierID=1").getResultList().get(0));
        product.setUnitsOnStock(product.getUnitsOnStock()-1);
        invoice.getProducts().add(product);
//        invoice.setQuantity(1);
//        em.persist(invoice);
        etx.begin();
        em.persist(invoice);
        etx.commit();
        em.close();

    }
}
```

```
INFO: HHH000115: Hibernate connection pool size: 20 (min=1)
maj 04, 2022 10:20:02 AM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
Hibernate: select supplier0_.SupplierID as supplier1_5_, supplier0_.City as city2_5_, supplier0_.CompanyName as companyn3_5_, supplier0_.Street as street4_5_ from Supplier supplier0
Hibernate: select next_val as id_val from hibernate_sequence
Hibernate: update hibernate_sequence set next_val= ? where next_val=?
Hibernate: select next_val as id_val from hibernate_sequence
Hibernate: update hibernate_sequence set next_val= ? where next_val=?
Hibernate: insert into Invoice (InvoiceNumber, Quantity, InvoiceID) values (?, ?, ?)
Hibernate: insert into Product (ProductName, UnitsOnStock, SupplierID, ProductID) values (?, ?, ?, ?)
Hibernate: insert into Invoice_Product (invoices_InvoiceID, products_ProductID) values (?, ?)

Process finished with exit code 0
```

	ProductID	ProductName	UnitsOnStock	SupplierID
1	1	Table	1	1
2	2	Chair	7	1
3	3	Drawer	13	1
4	7	testProduct	9	1

Tworzenie nowej faktury przy nowym produkcie.

```java
public class Main {
    public static void main(final String[] args)  {
        EntityManagerFactory emf = Persistence.createEntityManagerFactory( persistenceUnitName: "myDatabaseConfig");
        EntityManager em = emf.createEntityManager();
        EntityTransaction etx = em.getTransaction();
        Invoice invoice = new Invoice( invoiceNumber: "test3", quantity: 0);
//        Product product = (Product) em.createQuery("FROM Product WHERE ProductID=1").getResultList().get(0);
        Product product = new Product( productName: "testProduct3", unitsOnStock: 10);
        product.setSupplier((Supplier) em.createQuery( s: "FROM Supplier WHERE SupplierID=1").getResultList().get(0));
        product.setUnitsOnStock(product.getUnitsOnStock()-1);
        product.getInvoices().add(invoice);
        product.getInvoices().iterator().next().setQuantity(1);
//        invoice.getProducts().add(product);
//        invoice.setQuantity(1);
//        em.persist(invoice);
        em.persist(product);
        etx.begin();
        etx.commit();
        em.close();

    }
```

```
INFO: HHH000115: Hibernate connection pool size: 20 (min=1)
maj 04, 2022 10:53:03 AM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
Hibernate: select supplier0_.SupplierID as supplier1_5_, supplier0_.City as city2_5_, supplier0_.CompanyName as companyn3_5_, supplier0_.Street as street4_5_ from Supplier supplier0
Hibernate: select next_val as id_val from hibernate_sequence
Hibernate: update hibernate_sequence set next_val= ? where next_val=?
Hibernate: select next_val as id_val from hibernate_sequence
Hibernate: update hibernate_sequence set next_val= ? where next_val=?
Hibernate: insert into Product (ProductName, UnitsOnStock, SupplierID, ProductID) values (?, ?, ?, ?)
Hibernate: insert into Invoice (InvoiceNumber, Quantity, InvoiceID) values (?, ?, ?)

Process finished with exit code 0
```

```java
@ManyToMany(mappedBy = "products",cascade = CascadeType.PERSIST)
private Set<Invoice> invoices = new HashSet<>();
```

Zadanie 9

Przeniesienie adresów do embeded klasy

```java
@Embeddable
public class Address {
    private String Street;
    private String City;
//    @Id
//    @GeneratedValue(strategy = GenerationType.AUTO)
//    private int AddressID;

    public Address(String street, String city) {
        Street = street;
        City = city;
    }

    public String getStreet() {
        return Street;
    }

    public void setStreet(String street) {
        Street = street;
    }

    public String getCity() {
        return City;
    }

    public void setCity(String city) {
        City = city;
    }
```

```java
@Entity
public class Supplier {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private int SupplierID;
    private String CompanyName;
    private Address address;


    @OneToMany(mappedBy="supplier",cascade = CascadeType.PERSIST)
    private Set<Product> products = new HashSet<>();

    public Set<Product> getProducts() { return products; }

    public void setProducts(Set<Product> products) { this.products = products; }

    public Supplier() {
    }

    public Supplier(String companyName, String street, String city) {
        CompanyName = companyName;
        this.address = new Address(street,city);
    }

    public Address getAddress() {
        return address;
    }

    public void setAddress(Address address) {
        this.address = address;
    }

    public int getSupplierID() { return SupplierID; }

    public void setSupplierID(int supplierID) { SupplierID = supplierID; }
```

```
maj 04, 2022 11:17:51 AM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001001: Connection properties: {charSet=UTF-8, password=****, user=}
maj 04, 2022 11:17:51 AM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001003: Autocommit mode: false
maj 04, 2022 11:17:51 AM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledConnections <init>
INFO: HHH000115: Hibernate connection pool size: 20 (min=1)
maj 04, 2022 11:17:52 AM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]

Process finished with exit code 0
```

WHERE ORDER BY

	SupplierID	CompanyName	City	Street
1	1	OoohBi	Krakow	Krakowska

Przeniesienie adresów do osobnej tabeli

Stworzyła się taka tabela:

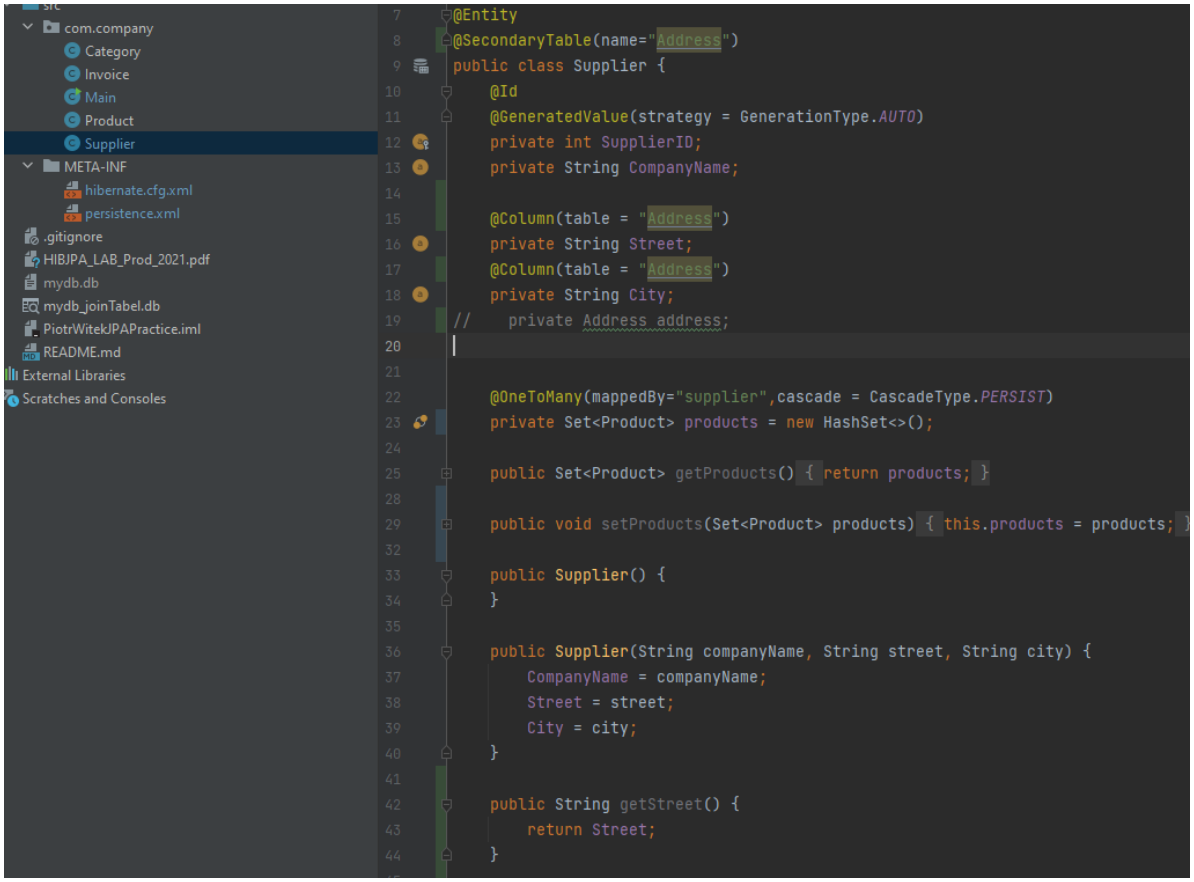
```
INFO: HHH10001003: Autocommit mode: false
maj 04, 2022 11:31:12 AM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledConnections <init>
INFO: HHH000115: Hibernate connection pool size: 20 (min=1)
Hibernate: create table HT_Supplier (SupplierID integer not null, hib_sess_id CHAR(36))
maj 04, 2022 11:31:12 AM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]

Process finished with exit code 0
```

WHERE ORDER BY

SupplierID	hib_sess_id

mydb.db 11
- main
  - tables 10
    - Address
    - Category
    - Category_Product
    - hibernate_sequence
    - HT_Supplier
    - Invoice
    - Invoice_Product
    - Product
    - sqlite_master
    - Supplier
  - Server Objects

```java
public class Main {
    public static void main(final String[] args)  {
        EntityManagerFactory emf = Persistence.createEntityManagerFactory( persistenceUnitName: "myDatabaseConfig");
        EntityManager em = emf.createEntityManager();
        EntityTransaction etx = em.getTransaction();
        Supplier supplier = new Supplier( companyName: "CEO", street: "Olszanska", city: "Krakow");
        em.persist(supplier);
        etx.begin();
        etx.commit();
        em.close();

    }
}
```



Dygresja:
Dopiero teraz doleciałem do slajdów o TypedQuery, a szkoda bo nie musiałbym tyle mapowania robić :(

Zadanie 10
Klasy dostawcy i klienta

```java
@Entity
public class Supplier extends Company{
    private String bankAccountNumber;


    @OneToMany(mappedBy="supplier",cascade = CascadeType.PERSIST)
    private Set<Product> products = new HashSet<>();

    public Set<Product> getProducts() { return products; }

    public void setProducts(Set<Product> products) { this.products = products; }

    public Supplier() {
    }

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

    public String getBankAccountNumber() { return bankAccountNumber; }

    public void setBankAccountNumber(String bankAccountNumber) { this.bankAccountNumber = bankAccountNumber; }
}
```

```java
@Entity
public class Customer extends Company{
    private String discount;

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

    public Customer() {
    }

    public String getDiscount() {
        return discount;
    }

    public void setDiscount(String discount) {
        this.discount = discount;
    }
}
```

Single_table

```java
@Entity
@SecondaryTable(name="Address")
@Inheritance(strategy = InheritanceType.SINGLE_TABLE)
public abstract class Company {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private int CompanyID;
    private String CompanyName;

    @Column(table = "Address")
    private String Street;
    @Column(table = "Address")
    private String City;
    @Column(table = "Address")
    private String ZipCode;

    public Company(String companyName, String street, String city, String zipCode) {
        CompanyName = companyName;
        Street = street;
        City = city;
        ZipCode = zipCode;
    }

    public Company() {
    }

    public int getCompanyID() {
        return CompanyID;
```

Dodawanie suppliera:

```java
public class Main {
    public static void main(final String[] args)  {
        EntityManagerFactory emf = Persistence.createEntityManagerFactory( persistenceUnitName: "myDatabaseConfig");
        EntityManager em = emf.createEntityManager();
        EntityTransaction etx = em.getTransaction();
        Supplier supplier = new Supplier( companyName: "alaMaKota", street: "Makowa", city: "Krakow", zipCode: "31-231", bankAccountNumber: "123455553434");
        em.persist(supplier);
        etx.begin();
        etx.commit();
        em.close();

    }
}
```

	DTYPE	CompanyID	CompanyName	discount	bankAccountNumber
1	Supplier	1	alaMaKota	<null>	123455553434

Dodawanie customera:

```
public class Main {
    public static void main(final String[] args)  {
        EntityManagerFactory emf = Persistence.createEntityManagerFactory( persistenceUnitName: "myDatabaseConfig");
        EntityManager em = emf.createEntityManager();
        EntityTransaction etx = em.getTransaction();
        Customer customer = new Customer( companyName: "alaMaKota", street: "Makowa", city: "Krakow", zipCode: "31-231",new BigDecimal( val: "22.13"));
        em.persist(customer);
        etx.begin();
        etx.commit();
        em.close();

    }
}
```

WHERE ORDER BY

	DTYPE	CompanyID	CompanyName	discount	bankAccountNumber
1	Supplier	1	alaMaKota	<null>	123455553434
2	Customer	2	alaMaKota	22.13	<null>

Joined

Jedyna zmiana w modelach to zamiana w Company.java

```
@Inheritance(strategy = InheritanceType.JOINED)
```

```
Hibernate: select next_val as id_val from hibernate_sequence
Hibernate: update hibernate_sequence set next_val= ? where next_val=?
Hibernate: select next_val as id_val from hibernate_sequence
Hibernate: update hibernate_sequence set next_val= ? where next_val=?
Hibernate: insert into Company (CompanyName, CompanyID) values (?, ?)
Hibernate: insert into Customer (discount, CompanyID) values (?, ?)
Hibernate: insert into Address (City, Street, ZipCode, CompanyID) values (?, ?, ?, ?)
Hibernate: insert into Company (CompanyName, CompanyID) values (?, ?)
Hibernate: insert into Supplier (bankAccountNumber, CompanyID) values (?, ?)
Hibernate: insert into Address (City, Street, ZipCode, CompanyID) values (?, ?, ?, ?)

Process finished with exit code 0
```

- Address
- Category
- Category_Product
- Company
- Customer
- hibernate_sequence
- HT_Company
- HT_Customer
- HT_Supplier
- Invoice
- Invoice_Product
- Product
- sqlite_master
- Supplier

	CompanyID	CompanyName
1	1	alaMaKota
2	2	alaNieMaKota

	bankAccountNumber	CompanyID
1	123455553434	2

discount	CompanyID
22.13	1

	City	Street	ZipCode	CompanyID
1	Krakow	Makowa	31-231	1
2	Krakow	Makowa	31-231	2

Table_per_class

Ponownie jedyna zmiana to

```
@Inheritance(strategy = InheritanceType.TABLE_PER_CLASS)
```

Wystąpiły problemy, które moim zdaniem są z winy SQLite'a, gdyż na podstawie:

5. Table per Class

The Table per Class strategy maps each entity to its table, which contains all the properties of the entity, including the ones inherited.

The resulting schema is similar to the one using @MappedSuperclass. But Table per Class will indeed define entities for parent classes, allowing associations and polymorphic queries as a result.

To use this strategy, we only need to add the *@Inheritance* annotation to the base class:

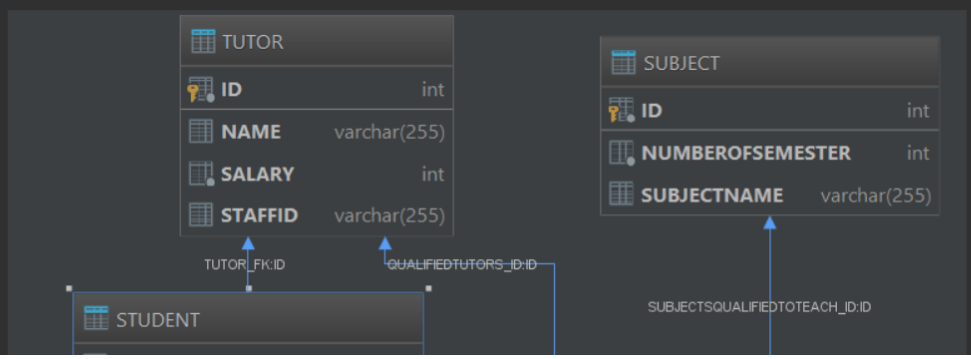
```
@Entity
@Inheritance(strategy = InheritanceType.TABLE_PER_CLASS)
public class Vehicle {
    @Id
    private long vehicleId;

    private String manufacturer;

    // standard constructor, getters, setters
}
```

Then we can create the subclasses in the standard way.

Oraz slajdu z wykładu

```
@Entity
@Inheritance(strategy= InheritanceType.TABLE_PER_CLASS)
public abstract class Person {
```

Jedna tabela na konkretną klasę



Zmieniłem tylko i wyłącznie ten fragment

```
@Inheritance(strategy = InheritanceType.TABLE_PER_CLASS)
```

W stosunku do poprzednich rodzajów dziedziczenia.

Otrzymałem poprawnie stworzony model:

```
Hibernate: drop table if exists Supplier
Hibernate: create table Address (City varchar(255), Street varchar(255), ZipCode varchar(255), CompanyID integer not null, primary key (CompanyID))
Hibernate: create table Category (CategoryID integer not null, Name varchar(255), primary key (CategoryID))
Hibernate: create table Category_Product (Category_CategoryID integer not null, Products_ProductID integer not null)
maj 04, 2022 1:26:14 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedConnection
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$ConnectionProviderJdbcConnectionAccess@1e1e9ef3] for (non-JTA) DDL execution was not in auto-com
Hibernate: create table Customer (CompanyID integer not null, CompanyName varchar(255), discount numeric(19, 2), primary key (CompanyID))
Hibernate: create table hibernate_sequence (next_val bigint)
Hibernate: insert into hibernate_sequence values ( 1 )
Hibernate: insert into hibernate_sequence values ( 1 )
Hibernate: insert into hibernate_sequence values ( 1 )
Hibernate: insert into hibernate_sequence values ( 1 )
Hibernate: create table Invoice (InvoiceID integer not null, InvoiceNumber varchar(255), Quantity integer not null, primary key (InvoiceID))
Hibernate: create table Invoice_Product (invoices_InvoiceID integer not null, products_ProductID integer not null, primary key (invoices_InvoiceID, products_ProductID))
Hibernate: create table Product (ProductID integer not null, ProductName varchar(255), UnitsOnStock integer, SupplierID integer not null, primary key (ProductID))
Hibernate: create table Supplier (CompanyID integer not null, CompanyName varchar(255), bankAccountNumber varchar(255), primary key (CompanyID))
Hibernate: alter table Category_Product add constraint UK_irycqnis1r42cijp8lnkuvlne unique (Products_ProductID)
maj 04, 2022 1:26:14 PM org.hibernate.tool.schema.internal.ExceptionHandlerLoggedImpl handleException
WARN: GenerationTarget encountered exception accepting command : Error executing DDL "alter table Category_Product add constraint UK_irycqnis1r42cijp8lnkuvlne unique (Products_ProductID)" via JDBC Statement
org.hibernate.tool.schema.spi.CommandAcceptanceException Create breakpoint : Error executing DDL "alter table Category_Product add constraint UK_irycqnis1r42cijp8lnkuvlne unique (Products_ProductID)" via JDBC Statement <13 internal
    at javax.persistence.Persistence.createEntityManagerFactory(Persistence.java:79)
    at javax.persistence.Persistence.createEntityManagerFactory(Persistence.java:54)
    at com.company.Main.main(Main.java:15)
Caused by: org.sqlite.SQLiteException Create breakpoint : [SQLITE_ERROR] SQL error or missing database (near "constraint": syntax error)
    at org.sqlite.core.DB.newSQLException(DB.java:941)
    at org.sqlite.core.DB.newSQLException(DB.java:953)
    at org.sqlite.core.DB.throwex(DB.java:918)
    at org.sqlite.core.NativeDB.prepare_utf8(Native Method)
    at org.sqlite.core.NativeDB.prepare(NativeDB.java:134)
    at org.sqlite.core.DB.prepare(DB.java:257)
    at org.sqlite.jdbc3.JDBC3Statement.execute(JDBC3Statement.java:52) <1 internal line>
    ... 15 more

maj 04, 2022 1:26:14 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]

Process finished with exit code 0
```

Wycinek i szkic

Wycinek został zapisany w schowku

Zaznacz tutaj, aby oznaczyć i udostępnić obraz

CompanyID	CompanyName	bankAccountNumber

CompanyID	CompanyName	discount

Ale przy insertowaniu modeli wychodzi zepsute mapowanie klas:

```
Hibernate: create table HT_Customer (CompanyID integer not null, hib_sess_id CHAR(36))
Hibernate: create table HT_Company (CompanyID integer not null, hib_sess_id CHAR(36))
Hibernate: create table HT_Supplier (CompanyID integer not null, hib_sess_id CHAR(36))
maj 04, 2022 1:28:34 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
Hibernate: select next_val as id_val from hibernate_sequence
Hibernate: update hibernate_sequence set next_val= ? where next_val=?
Hibernate: select next_val as id_val from hibernate_sequence
Hibernate: update hibernate_sequence set next_val= ? where next_val=?
Hibernate: insert into Customer (CompanyName, City, Street, ZipCode, discount, CompanyID) values (?, ?, ?, ?, ?, ?)
maj 04, 2022 1:28:35 PM org.hibernate.engine.jdbc.spi.SqlExceptionHelper logExceptions
WARN: SQL Error: 1, SQLState: null
maj 04, 2022 1:28:35 PM org.hibernate.engine.jdbc.spi.SqlExceptionHelper logExceptions
ERROR: [SQLITE_ERROR] SQL error or missing database (table Customer has no column named City)
Exception in thread "main" javax.persistence.RollbackException Create breakpoint : Error while committing the transaction <2 internal lines>
    at com.company.Main.main(Main.java:23)
Caused by: javax.persistence.PersistenceException Create breakpoint : org.hibernate.exception.GenericJDBCException: could not prepare statement <3 internal line
    ... 2 more
Caused by: org.hibernate.exception.GenericJDBCException Create breakpoint : could not prepare statement <10 internal lines>
    at java.base/java.util.LinkedHashMap.forEach(LinkedHashMap.java:684) <13 internal lines>
    ... 1 more
Caused by: org.sqlite.SQLiteException Create breakpoint : [SQLITE_ERROR] SQL error or missing database (table Customer has no column named City)
    at org.sqlite.core.DB.newSQLException(DB.java:941)
    at org.sqlite.core.DB.newSQLException(DB.java:953)
    at org.sqlite.core.DB.throwex(DB.java:918)
    at org.sqlite.core.NativeDB.prepare_utf8(Native Method)
    at org.sqlite.core.NativeDB.prepare(NativeDB.java:134)
    at org.sqlite.core.DB.prepare(DB.java:257)
    at org.sqlite.core.CorePreparedStatement.<init>(CorePreparedStatement.java:47)
    at org.sqlite.jdbc3.JDBC3PreparedStatement.<init>(JDBC3PreparedStatement.java:30)
    at org.sqlite.jdbc4.JDBC4PreparedStatement.<init>(JDBC4PreparedStatement.java:19)
    at org.sqlite.jdbc4.JDBC4Connection.prepareStatement(JDBC4Connection.java:35)
    at org.sqlite.jdbc3.JDBC3Connection.prepareStatement(JDBC3Connection.java:241)
    at org.sqlite.jdbc3.JDBC3Connection.prepareStatement(JDBC3Connection.java:205) <2 internal lines>
    ... 21 more

Process finished with exit code 1
```

Przeszukiwanie sieci nie rozwiązało problemu. Bardzo zależy mi na tym, żeby się dowiedzieć dlaczego to nie działa.

W załączniku obok przezyłam zzipowany kod projektu.