

Hibernate, JPA – laboratorium

I. Basics

- a. Stwórz folder INazwiskoJPAPractice. Aby uniknąć na późniejszym etapie problemów z uprawnieniami najbezpieczniej założyć ten folder w katalogach „ogólnie dostępnych” np. /tmp lub /mnt/vm/własne-kopie
- b. Ściągnij i rozpakuj w stworzonym folderze DBMS Derby <https://www-eu.apache.org/dist/db/derby/db-derby-10.14.2.0/db-derby-10.14.2.0-bin.zip>
- c. Uruchom serwer Derby (skrypt startNetworkServer z podkatalogu bin)

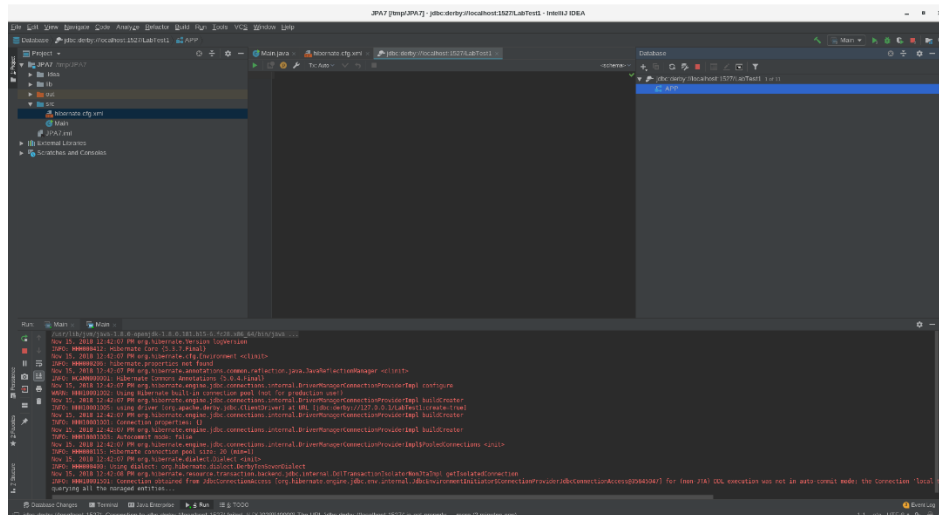
- d. Stwórz projekt Javowy o nazwie InazwiskoJPAPractice (IntelliJ, Eclipse wszystko jedno, choć najprościej chyba zakładać projekt w IntelliJ a w kreatorze projektu zaznaczyć dołączenie Hibernate’a (i od razu zaznaczyć opcje wygenerowania pliku konfiguracyjnego hibernate’a i maina).

BARDZO WAŻNE!

Zwróć uwagę projekt używał JDK w wersji nie wyższej niż 1.9 (w praktyce 1.8)

Żeby uniknąć problemów z dostępem do projektu/bibliotek załóż projekt w katalogu tworzonym na początku zajęć

- e. Dołącz do projektu (w IJ: File→Project Structure → Modules→ Dependencies) Jar-ki Związane z obsługą/komunikacją z Derby (derby.jar, derbyclient.jar, derbynet.jar, derbytools.jar). Znajdziesz je w podkatalogu lib ściągniętego Derby.
- f. Uzupełnij wpisy w hibernate.cfg.xml podając driver, connection_url, dodaj opcje show_sql oraz format_sql, a także hbm2ddl na create. Przed pierwszym uruchomieniem connection_url powinien wyglądać jak poniżej
jdbc:derby://127.0.0.1/INazwiskoJPA;create=true'; (z dokładnością do nazwy bazy danych). Po pierwszym uruchomieniu usuń dopisek create=true.
- g. Uruchom projekt, Na razie nie będzie się wiele działo ale na konsoli powinieneś zobaczyć wpisy hibernate, bez żadnych wyjątków, czyli mniej więcej stan jak poniżej:



- Ewentualnie możesz uruchomić shell'a serwera derby (narzędzie o nazwie ij z podkatalogu bin Derby)
- Po uruchomieniu ij podepnij się do serwera czyli polecenie
connect 'jdbc:derby://127.0.0.1/INazwiskoJPA';
- I następnie polecenie show tables; na ij powinno pokazac Ci serie tabel systemowych, czyli cos takiego jak poniżej:



- Wracamy do IntelliJ'a
- Stwórz klasę produktu z polami ProductName, UnitsOnStock
- Zapewnij Mapowanie klasy do tabeli Products
- Uzupełnij w projekcie elementy potrzebne do zmapowania klasy do bazy danych (jeżeli nie wygenerowałeś na etapie zakładania projektu - szkielet hibconfiga poniżej):

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
    <session-factory>
        <property name="connection.driver_class">org.apache.derby.jdbc.ClientDriver</property>
        <property name="connection.url">jdbc:derby://localhost/MyDatabase</property>
        <property name="show_sql">true</property>
        <property name="format_sql">true</property>
        <property name="use_sql_comments">true</property>
        <property name="hbm2ddl.auto">create-drop</property>
    </session-factory>
</hibernate-configuration>
```

- h. Jeżeli nie wygenerowałeś maina z tworzeniem sesji na etapie zakładania projektu w pliku głównym zapewnij tworzenie sesji i jej otwieranie. Poniżej przykładowa implementacja:

```
public class Main {

    private static SessionFactory sessionFactory = null;

    public static void main(String[] args) {
        sessionFactory = getSessionFactory();
        Session session = sessionFactory.openSession();
        Transaction tx = session.beginTransaction();
        tx.commit();
        session.close();
    }

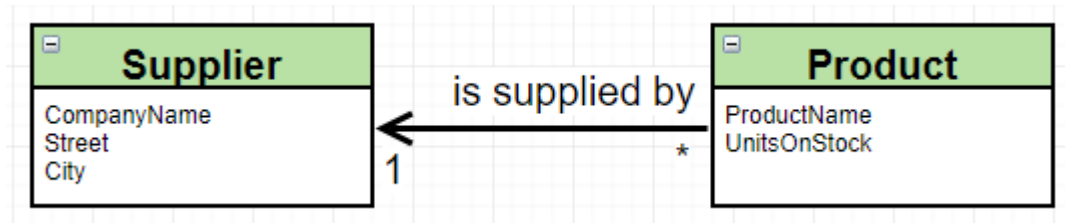
    private static SessionFactory getSessionFactory() {
        if (sessionFactory == null) {
            Configuration configuration = new Configuration();
            sessionFactory = configuration.configure().buildSessionFactory();
        }
        return sessionFactory;
    }
}
```

- i. Zapytaj użytkownika o nazwę produktu i jego aktualny stan magazynowy.

```
II. Scanner inputScanner = new Scanner(System.in);
    String prodName = inputScanner.nextLine();
```

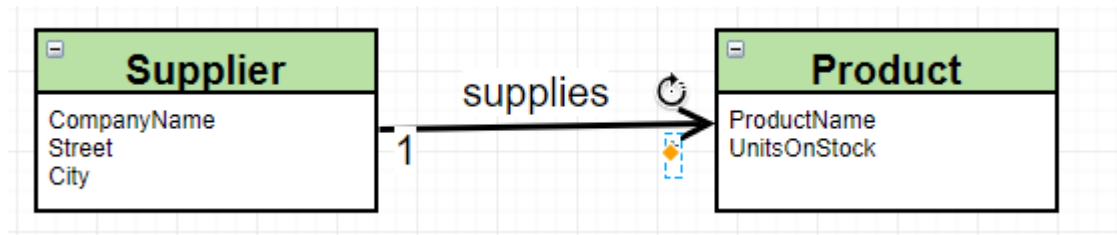
- a. Stwórz produkt o podanej nazwie i stanie magazynowym i utrwal go w BD z wykorzystaniem Hibernate'a.
- b. **Udokumentuj wykonane kroki oraz uzyskany rezultat (ogi wywołań sqlowych, describe table/diagram z datagrip, select * from....)**

III. Zmodyfikuj model wprowadzając pojęcie Dostawcy jak poniżej



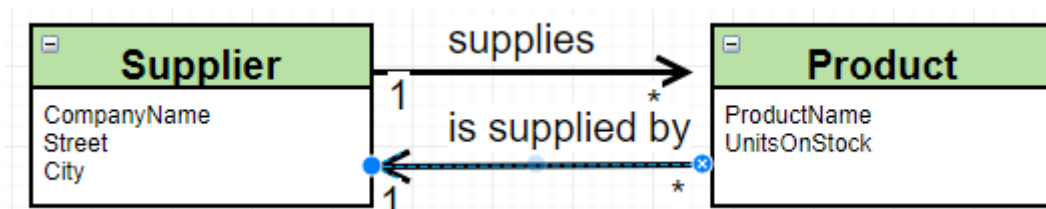
- a. Stwórz nowego dostawcę.
- b. Znajdź poprzednio wprowadzony produkt i ustaw jego dostawcę na właśnie dodanego.
- c. **Udokumentuj wykonane kroki oraz uzyskany rezultat (ogi wywołań sqlowych, describe table/diagram z datagrip, select * from....)**

IV. Odwróć relacje zgodnie z poniższym schematem



- a. Zamodeluj powyższe w dwóch wariantach „z” i „bez” tabeli łącznikowej
- b. Stwórz kilka produktów
- c. Dodaj je do produktów dostarczanych przez nowo stworzonego dostawcę
- d. **Udokumentuj wykonane kroki oraz uzyskane rezultaty w obu wariantach (logi wywołań sqlowych, describe table/diagram z datagrip, select * from....)**

V. Zamodeluj relacje dwustronną jak poniżej:



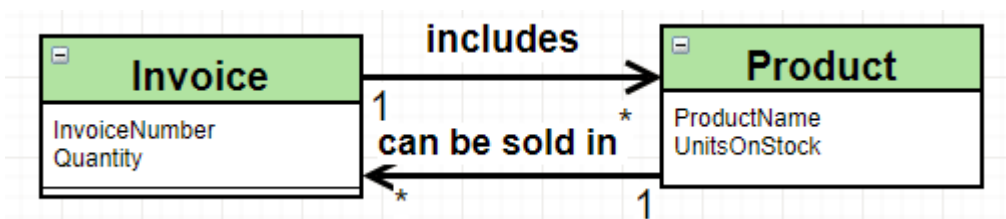
- a. Tradycyjnie: Stwórz kilka produktów
- b. Dodaj je do produktów dostarczanych przez nowo stworzonego dostawcę (pamiętaj o poprawnej obsłudze dwustronności relacji)

- c. Udokumentuj wykonane kroki oraz uzyskane rezultaty (logi wywołań sqlowych, describe table/diagram z datagrip, select * from....)

VI. Dodaj klasę Category z property int CategoryID, String Name oraz listą produktów List<Product> Products

- Zmodyfikuj produkty dodając wskazanie na kategorii do której należy.
- Stworz kilka produktow i kilka kategorii
- Dodaj kilka produktów do wybranej kategorii
- Wydobądź produkty z wybranej kategorii oraz kategorię do której należy wybrany produkt
- Udokumentuj wykonane kroki oraz uzyskane rezultaty (logi wywołań sqlowych, describe table/diagram z datagrip, select * from....)

VII. Zamodeluj relacje wiele-do-wielu, jak poniżej:



- Stórz kilka produktów i “sprzedaj” je na kilku transakcjach.
- Pokaż produkty sprzedane w ramach wybranej faktury/transakcji
- Pokaż faktury w ramach których był sprzedany wybrany produkt
- Udokumentuj wykonane kroki oraz uzyskane rezultaty (logi wywołań sqlowych, describe table/diagram z datagrip, select * from....)

VIII. Kontynuuj jeśli masz czas lub zrealizuj jako zadanie domowe

IX. JPA

- a. Stwórz nowego maina w którym zrobisz to samo co w punkcie VI ale z wykorzystaniem JPA
- b. **Udokumentuj wykonane kroki oraz uzyskane rezultaty (logi wywołań sqlowych, describe table/diagram z datagrip, select * from....)**

X. Kaskady

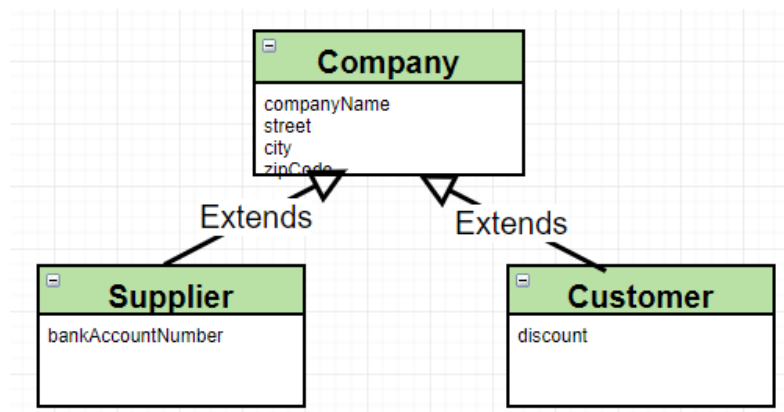
- a. Zmodyfikuj model w taki sposób aby było możliwe kaskadowe tworzenie faktur wraz z nowymi produktami, oraz produktów wraz z nową fakturą
- b. **Udokumentuj wykonane kroki oraz uzyskane rezultaty (logi wywołań sqlowych, describe table/diagram z datagrip, select * from....)**

XI. Embedded class

- a. Dodaj do modelu klase adres. „Wbuduj” ją do tabeli Dostawców.
- b. **Udokumentuj wykonane kroki oraz uzyskane rezultaty (logi wywołań sqlowych, describe table/diagram z datagrip, select * from....)**
- c. Zmodyfikuj model w taki sposób, że dane adresowe znajdują się w klasie dostawców. Zmapuj to do dwóch osobnych tabel.
- d. **Udokumentuj wykonane kroki oraz uzyskane rezultaty (logi wywołań sqlowych, describe table/diagram z datagrip, select * from....)**

XII. Dziedziczenie

- a. Wprowadź do modelu następującą hierarchie:



- b. Dodaj i pobierz z bazy kilka firm obu rodzajów stosując po kolei trzy różne strategie mapowania dziedziczenia.
- c. **Udokumentuj wykonane kroki oraz uzyskane rezultaty (logi wywołań sqlowych, describe table/diagram z datagrip, select * from....)**

XIII. Aplikacja do zamawiania produktów

- a. Rozszerz projekt/zaimplementuj aplikację wprowadzając możliwość składania zamówień na produkty (ew inne funkcjonalności wg własnego pomysłu stosownie do umiejętności/czasu/preferencji/chęci poznania).
- b. Zwróć szczególną uwagę na wykorzystanie mechanizmów Hibernate/JPA
- c. Zadanie (stosownie do umiejętności/czasu/preferencji/chęci poznania) może zostać zrealizowane jako:
 - i. Rozszerzenie projektu rozpoczętego na zajęciach
 - ii. Aplikacja „okienkowa”
 - iii. Aplikacja webowa
- d. Przygotuj sprawozdanie omawiające przygotowaną aplikację, wykorzystane mechanizmy i sposób ich działania (wraz ze screenami).