

Hibernate, JPA – laboratorium

- I. Z uwagi na tryb prowadzenia zajęć na bieżąco dokumentuj wykonywane operacje w dokumencie INazwiskoJPALab. Na koniec zajęć – niezależnie od etapu na którym będziesz wrzuć to sprawozdanie do zadania EfektyPracyNaKoniecZajęć na moodlu.

II. Basics

- a. Ściągnij i rozpakuj DBMS Derby <https://downloads.apache.org/db/derby/db-derby-10.15.2.0/db-derby-10.15.2.0-bin.zip>
- b. Uruchom serwer Derby (skrypt startNetworkServer z podkatalogu bin) – najlepiej wejść do tego katalogu z terminala i wywołać „z palca” ./startNetworkServer – jak coś będzie nie tak to zobaczysz przynajmniej komunikaty błędów
- c. Derby 10.15 działa od Javy 9. Jeśli masz jakąś starszą co objawi się podczas uruchamiania serwera derby błędem typu:
- d. Exception in thread "main" java.lang.UnsupportedClassVersionError: org/apache/derby/drda/NetworkServerControl has been compiled by a more recent version of the Java Runtime (class file version 53.0), this version of the Java Runtime only recognizes class file versions up to 52.0
- e. – aktualizuj Javę, albo przestaw JAVA_HOME na nowszą jeśli już masz albo ściągnij starsze DerbyDB (10.14) (jeśli już coś musisz ściągać to ściągnięcie starszego derby będzie szybsze ☺)
- f. Uruchom konsolę ij z podkatalogu bin Derby (ew podłącz się pod uruchomiony serwer z poziomu DataGrip czy IntelliJ jeśli wolisz)
- g. Połącz się do serwera zakładając bazę INazwiskoJPA. Czyli w kliencie ij polecenie `connect 'jdbc:derby://127.0.0.1/INazwiskoJPA;create=true';`
(create=true wywołaj tylko raz)
- h. polecenie `show tables;` na ij powinno pokazać Ci serie tabel systemowych, czyli coś takiego jak poniżej:

```
Terminal
SYS      |SYSCOLPERMS |
SYS      |SYSCOLUMNS|
SYS      |SYSCONGLOMERATES|
SYS      |SYSCONSTRAINTS|
SYS      |SYSDEPENDS  |
SYS      |SYSFILES    |
SYS      |SYSFORIGNKEYS|
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 rows selected
ij>
```

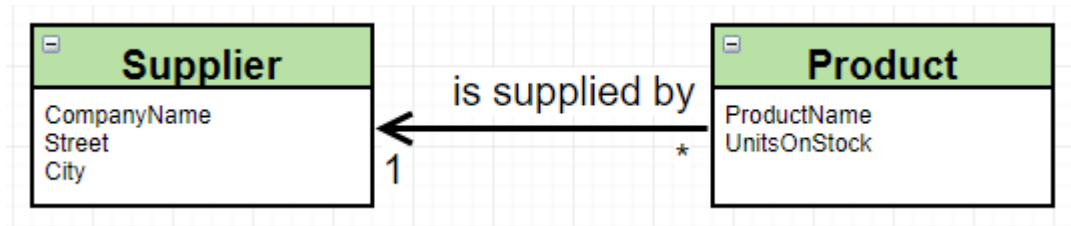
- i. Stwórz projekt Javowy o nazwie InazwiskoJPAPractice (IntelliJ, Eclipse wszystko jedno, choć najprościej chyba zakładać projekt w IntelliJ a w kreatorze projektu zaznaczyć dołączenie Hibernate'a i zaznaczyć tworzenie maina i configa).
- j. Stwórz klasę produktu z polami ProductName, UnitsOnStock
- k. Uzupełnij w klasie elementy potrzebne do zmapowania klasy do bazy danych
- l. Dołącz do projektu (w IntelliJ: File→Project Structure → Modules→ Dependencies) Jar-ki Związane z obsługą/komunikacją z Derby (derbyclient.jar, derbytools.jar). Znajdziesz je w podkatalogu lib ściągniętego Derby.
- m. Uzupełnij potrzebne property w konfiguracji hibernate'a (szkielet hibconfiga poniżej):

```
III. <?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="connec-
tion.url">jdbc:derby://127.0.0.1/INazwiskoJPA</property>
      <property name="connec-
tion.driver_class">org.apache.derby.jdbc.ClientDriver</property>
      <property name="dialect">org.hibernate.dialect.DerbyTenSe-
venDialect</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>
```

- a. W mainie stwórz przykładowy produkt i utwórz go w BD z wykorzystaniem Hibernate'a.

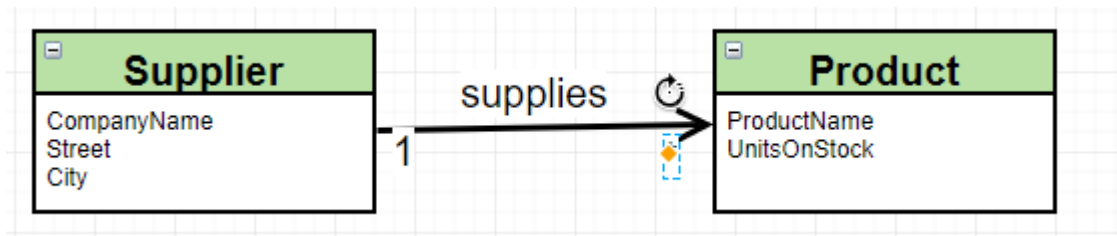
- b. Udokumentuj wykonane kroki oraz uzyskany rezultat (logi wywołań słowych, describe table/diagram z datagrip, select * from....)

IV. Zmodyfikuj model wprowadzając pojęcie Dostawcy jak poniżej



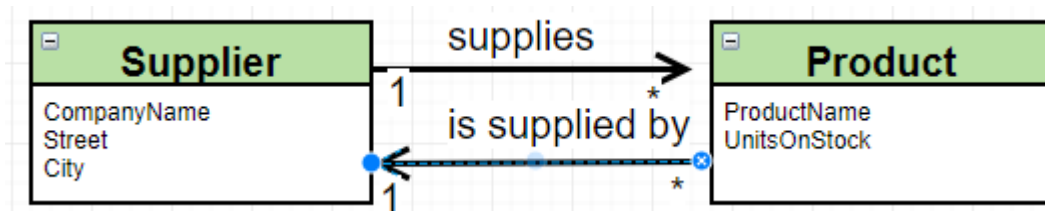
- a. Stworz nowego dostawce.
- b. Znajdz poprzednio wprowadzony produkt i ustaw jego dostawce na właśnie dodanego.
- c. Udokumentuj wykonane kroki oraz uzyskany rezultat (logi wywołań słowych, describe table/diagram z datagrip, select * from....)

V. Odwróć relacje zgodnie z poniższym schematem



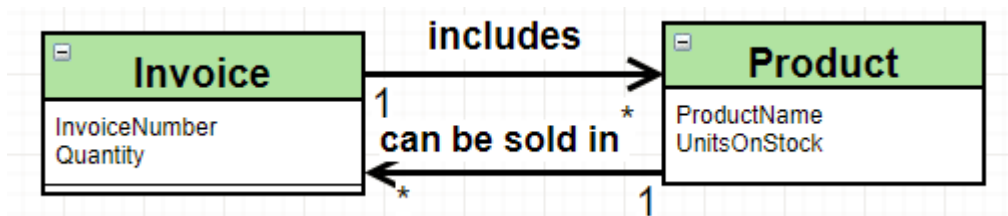
- a. Zamodeluj powyższe w dwóch wariantach „z” i „bez” tabeli łącznikowej
- b. Stworz kilka produktow
- c. Dodaj je do produktow dostarczanych przez nowo stworzonego dostawcę
- d. Udokumentuj wykonane kroki oraz uzyskane rezultaty w obu wariantach (logi wywołań słowych, describe table/diagram z datagrip,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

VI. Zamodeluj relacje dwustronną jak poniżej:



- a. Tradycyjnie: Stworz kilka produktow
- b. Dodaj je do produktow dostarczanych przez nowo stworzonego dostawcę (pamiętaj o poprawnej obsłudze dwustronności relacji)
- c. Udokumentuj wykonane kroki oraz uzyskane rezultaty (logi wywołań słowych, describe table/diagram z datagrip, select * from....)

- VII. 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....)**
- VIII. 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....)**

IX. Kontynuuj jeśli masz czas lub zrealizuj jako zadanie domowe

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

XI. Kaskady

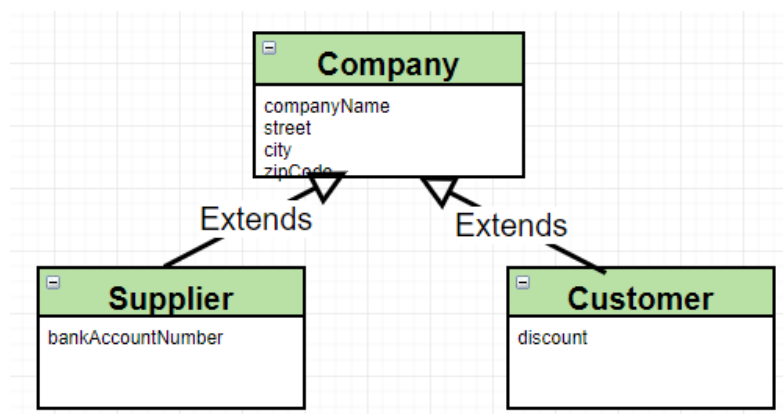
- a. Zmodyfikuj model w taki sposób aby było możliwe kaskadowe tworzenie faktur wraz z nowymi produktami, oraz produktów wraz z nową fakturą
- b. **Udokumentuj wykonane kroki oraz uzyskane rezultaty (logi wywołań sglowych, describe table/diagram z datagrip, select * from....)**

XII. Embedded class

- a. Dodaj do modelu klase adres. „Wbuduj” ją do tabeli Dostawców.
- b. **Udokumentuj wykonane kroki oraz uzyskane rezultaty (logi wywołań sglowych, 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 sglowych, describe table/diagram z datagrip, select * from....)**

XIII. 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 sglowych, describe table/diagram z datagrip, select * from....)**