



# Jak správně na Hibernate

Tomáš Krátký  
Martin Hlavatý

<http://www.profinit.eu/>

<http://www.profinit.eu/cz/podpora-univerzit>



# Agenda

- Základní koncepty ORM
- Plain Old JDBC
- Základy Hibernate
- Hibernate klasicky (webové aplikace)
- Hibernate neklasicky (desktop aplikace)
- iBatis (konkurence Hibernate)
- Migrace DB platformy
- Závěr, diskuse





# Úvod

- Co je Hibernate?

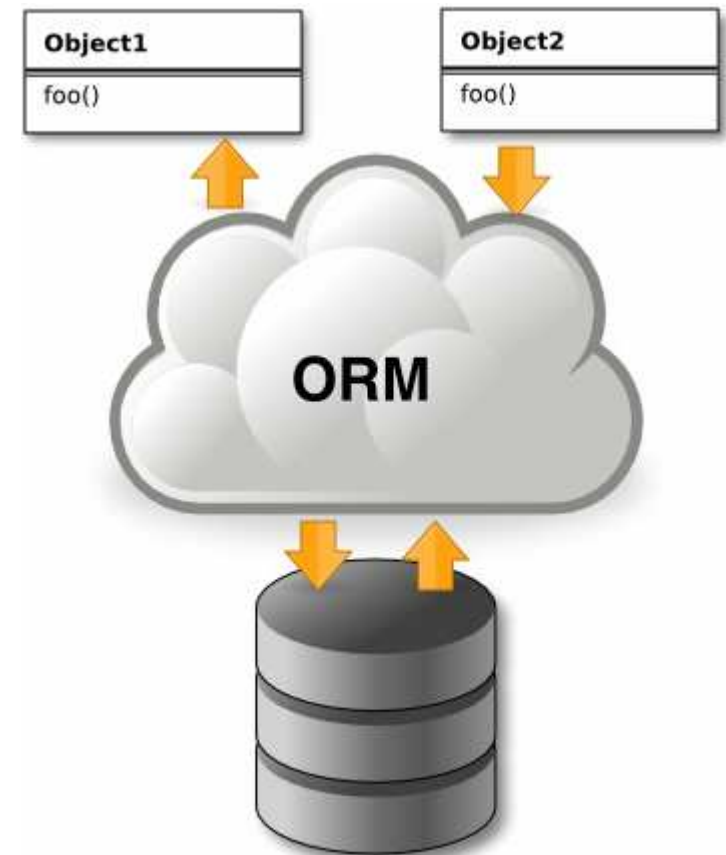


- Hibernate je ORM!



# Úvod

- ORM = objektově relační mapování
- Konverze mezi relačním a objektovým světem
  - Specifikuje se pomocí metadat (XML, anotace, ...)
- Efekt virtuální objektové databáze





# ORM - Motivace

- Odstranění duplikace kódu při práci s databází
  - Vyšší produktivita vývojářů
  - Méně kódu = méně chyb
- Udržovatelnost – snazší provádění změn
  - Doménový a datový model jsou svázány pomocí metadat
- Nezávislost na konkrétním RDBMS
  - Bohužel v praxi je to nereálné



# Trocha historie

- CORBA Component Model
  - Součást CORBA 3.0 (rok 2002)
  - Kategorie **entity** komponent
- Enterprise Java Beans
  - **Entity** beans (Container Managed Persistence)
  - První verze (1998 – 2000) prakticky nepoužitelné
  - Od verze 2.x (rok 2001) použitelné ale neobratné
  - Od verze 3.0 (rok 2006) zrušeny a nahrazeny JPA

# EJB vs. JPA vs. Hibernate

- JPA = Java Persistence API
- JPA 1.0 součástí standardu EJB 3.0
- Hibernate implementuje JPA
- JPA vzniklo z Hibernate (a dalších)







# Impedanční neshoda

## Rozdíl v paradigmatech světa relačního a objektového

### Problém granularity

- Kompozice objektů → složitější objekty. U relací nelze.

### Dědičnost

- Relační teorie tento pojem nezná

### Problém identity

- V Javě equals a ==, u relací primární klíč

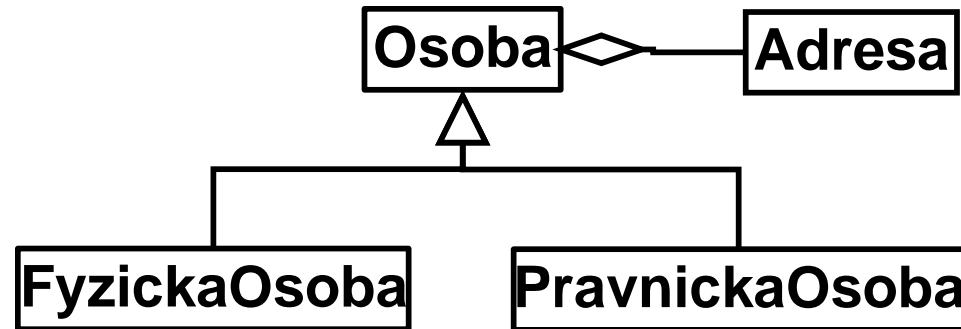
### Problém asociací

- Cizí klíče jsou jednosměrné





# Příklad



- V databázi může být uloženo ve dvou, třech či čtyřech tabulkách.
  - V případě tří tabulek je problém s vazbou adresy
    - Nelze použít cizí klíč (neexistuje tabulka reprezentující předka).



# JDBC





# JDBC

- Standardní způsob práce s DB v Javě
- Prakticky všechny ORM jej používají
- Výhody
  - Jednoduché
  - Plná kontrola nad SQL
  - Dobrá podpora uložených procedur
- Nevýhody
  - Checked výjimky
  - Mnoho opakujícího se kódu



# JDBC - příklad

```
Connection con = null; Statement stmt = null; ResultSet rs = null;
try {
    con = getConnection();
    stmt = con.createStatement();
    rs = stmt.executeQuery("SELECT jmeno, prijmeni FROM osoby");
    while (rs.next()) {
        Osoba osoba = new Osoba();
        osoba.setJmeno( rs.getString("jmeno") );
        osoba.setPrijmeni(rs.getString("prijmeni "));
        osoby.add(osoba);
    }
} catch (Exception e) {
    ...
} finally {
    if ( rs != null ) try { rs.close(); } catch (SQLException e) {}
    rs = null;
    if ( stmt != null ) try { stmt.close(); } catch (SQLException e) {}
    stmt = null;
    if ( con != null ) try { con.close(); } catch (SQLException con) {}
    con = null;
}
```



# JDBC - příklad

```
Connection con = null; Statement stmt = null; ResultSet rs = null;
try {
    con = getConnection();
    stmt = con.createStatement();
    rs = stmt.executeQuery("SELECT jmeno, prijmeni FROM osoby");
    while (rs.next()) {
        Osoba osoba = new Osoba();
        osoba.setJmeno( rs.getString("jmeno") );
        osoba.setPrijmeni(rs.getString(" prijmeni "));
        osoby.add(osoba);
    }
} catch (Exception e) {
    ...
} finally {
    if ( rs != null ) try { rs.close(); } catch (SQLException e) {}
    rs = null;
    if ( stmt != null ) try { stmt.close(); } catch (SQLException e) {}
    stmt = null;
    if ( con != null ) try { con.close(); } catch (SQLException con) {}
    con = null;
}
```



# JDBC - příklad

```
Connection con = null; Statement stmt = null; ResultSet rs = null;
try {
    con = getConnection();
    stmt = con.createStatement();
    rs = stmt.executeQuery("SELECT jmeno, prijmeni FROM osoby");
    while (rs.next()) {
        Osoba osoba = new Osoba();
        osoba.setJmeno( rs.getString("jmeno") );
        osoba.setPrijmeni(rs.getString(" prijmeni "));
        osoby.add(osoba);
    }
} catch (Exception e) {
    ...
} finally {
    if ( rs != null ) try { rs.close(); } catch (SQLException e) {}
    rs = null;
    if ( stmt != null ) try { stmt.close(); } catch (SQLException e) {}
    stmt = null;
    if ( con != null ) try { con.close(); } catch (SQLException con) {}
    con = null;
}
```



# JDBC - příklad

```
Connection con = null; Statement stmt = null; ResultSet rs = null;
try {
    con = getConnection();
    stmt = con.createStatement();
    rs = stmt.executeQuery("SELECT jmeno, prijmeni FROM osoby");
    while (rs.next()) {
        Osoba osoba = new Osoba();
        osoba.setJmeno( rs.getString("jmeno") );
        osoba.setPrijmeni(rs.getString(" prijmeni " ) );
        osoby.add(osoba);
    }
} catch (Exception e) {
    ...
} finally {
    if ( rs != null ) try { rs.close(); } catch (SQLException e) {}
    rs = null;
    if ( stmt != null ) try { stmt.close(); } catch (SQLException e) {}
    stmt = null;
    if ( con != null ) try { con.close(); } catch (SQLException con) {}
    con = null;
}
```







# Vylepšení

- **Template method**
  - Získání Connection
  - Ošetření výjimek
- **RowMapper**
  - ResultSet → doménový objekt





# JDBC a Spring

- JdbcTemplate
  - Konverze výjimek na unchecked
  - Správa Connections
  - Podpora deklarativních transakcí
  - RowMappers
- Výhody
  - Jednoduché
  - K dispozici plná síla JDBC
- Nevýhody
  - Stále poměrně pracné





# Příklad

```
public Collection getOsoby() {  
    return getJdbcTemplate().query(  
        "select jmeno, prijmeni from osoby", new OsobaMapper()  
    );  
}
```

```
static class OsobaMapper implements RowMapper {  
    public Object mapRow(ResultSet rs, int rowNum)  
        throws SQLException {  
        Osoba osoba = new Osoba();  
        osoba.setJmeno( rs.getString("jmeno") );  
        osoba.setPrijmeni( rs.getString("prijmeni") );  
        return osoba;  
    }  
}
```



# Příklad

```
public Collection getOsoby() {  
    return getJdbcTemplate().query(  
        "select jmeno, prijmeni from osoby", new OsobaMapper()  
    );  
}
```

```
static class OsobaMapper implements RowMapper {  
    public Object mapRow(ResultSet rs, int rowNum)  
        throws SQLException {  
        Osoba osoba = new Osoba();  
        osoba.setJmeno( rs.getString("jmeno") );  
        osoba.setPrijmeni( rs.getString("prijmeni") );  
        return osoba;  
    }  
}
```



# Příklad

```
public Collection getOsoby() {  
    return getJdbcTemplate().query(  
        "select jmeno, prijmeni from osoby", new OsobaMapper()  
    );  
}
```

```
static class OsobaMapper implements RowMapper {  
    public Object mapRow(ResultSet rs, int rowNum)  
    throws SQLException {  
        Osoba osoba = new Osoba();  
        osoba.setJmeno( rs.getString("jmeno") );  
        osoba.setPrijmeni( rs.getString("prijmeni") );  
        return osoba;  
    }  
}
```



# Hibernate







# Proč Hibernate?





# Vyspělá technologie



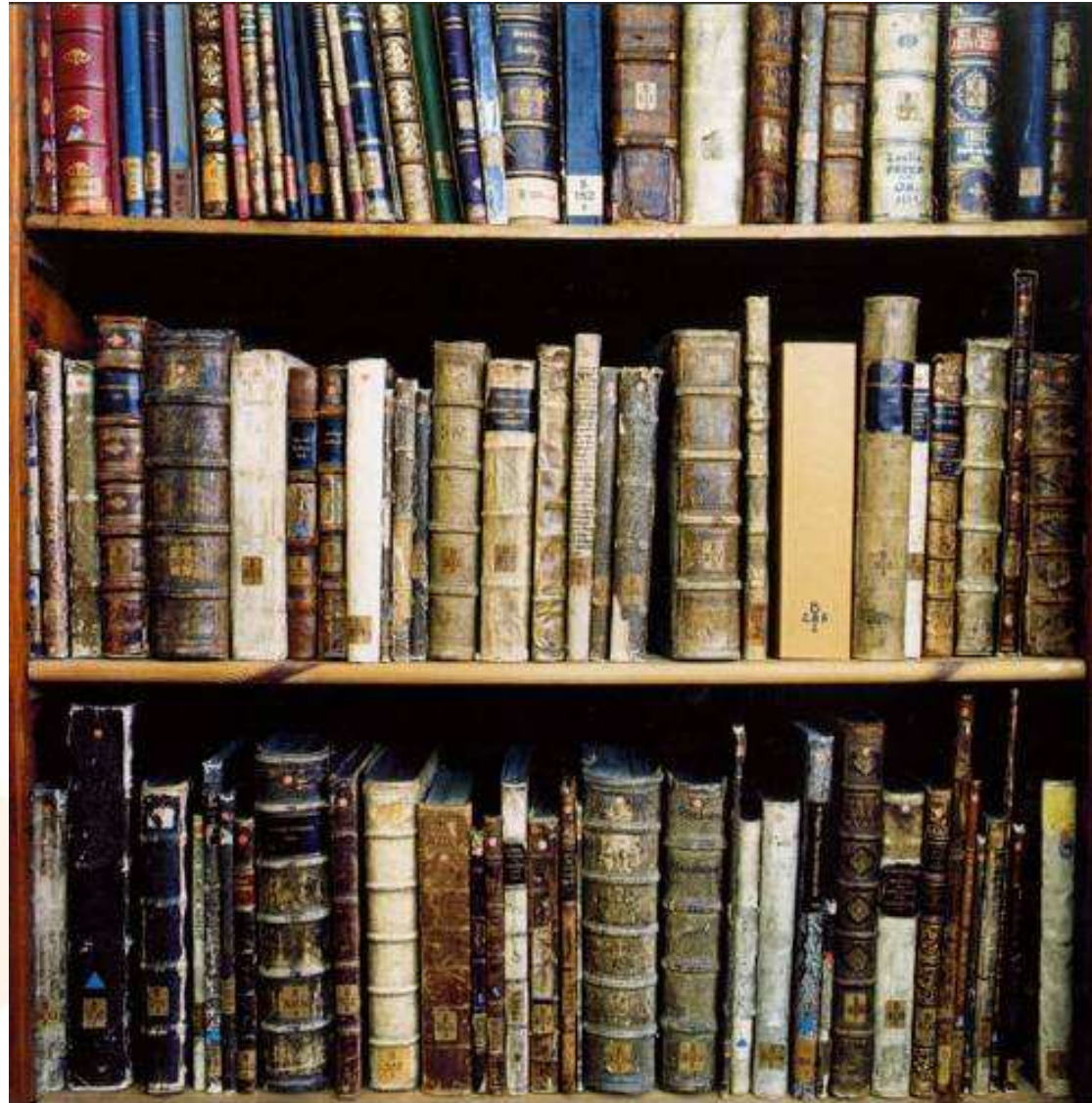


# Velká komunita





# Dostatečná dokumentace







# Dobrá architektura





# Proč Hibernate?

- Mnoho přidružených projektů
  - Validece – Hibernate Validator
  - Fulltextové vyhledávání – Hibernate Search
  - Horizontální partitioning – Hibernate Shards
- Možnost „*kdykoliv*“ změnit použitou databázi
  - U větších projektů je tato možnost spíše teoretická
  - Podpora velkého množství databázových serverů

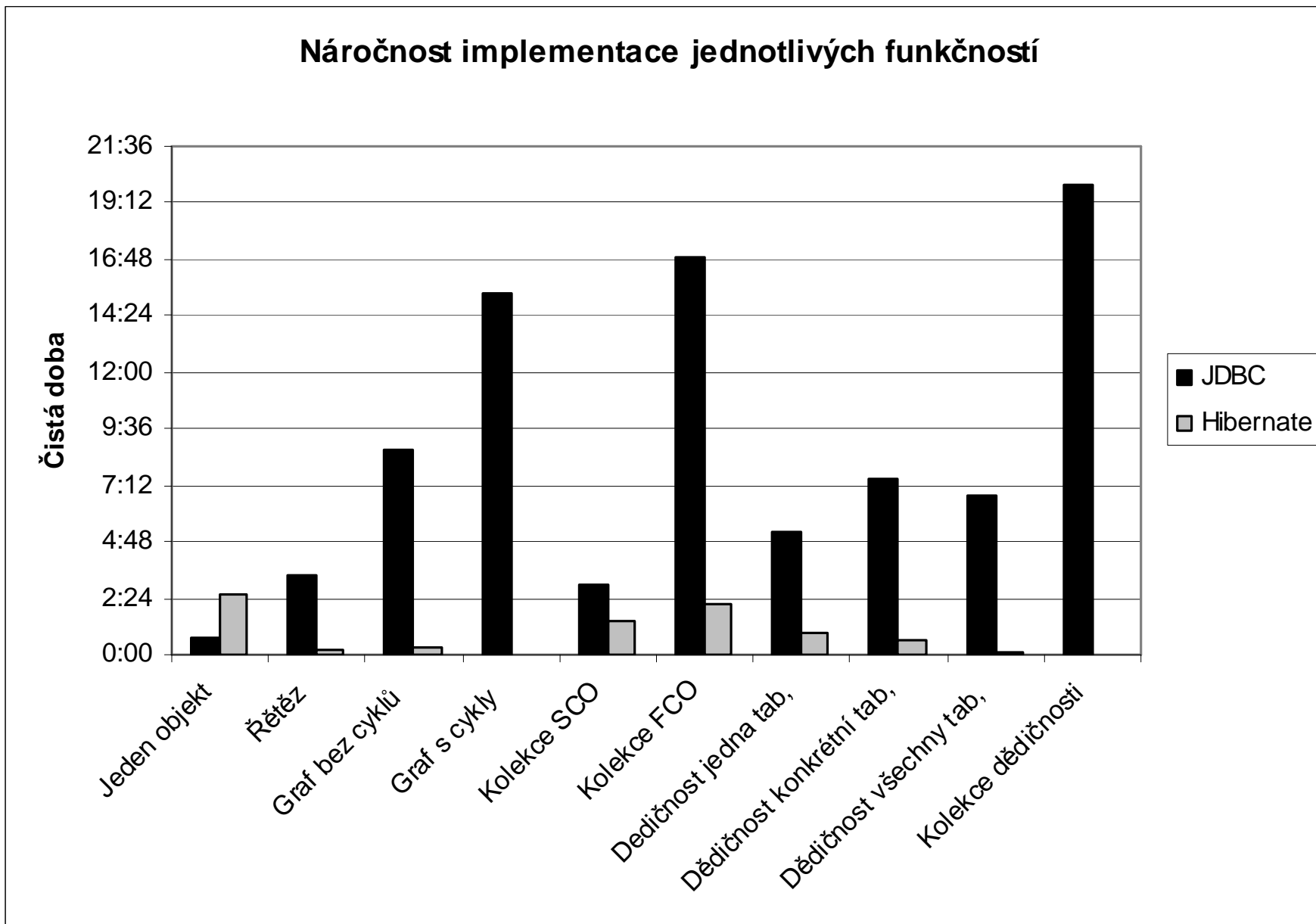


# Proč Hibernate?

- Pracuje s POJOs
- API pro základní CRUD operace
  - Není třeba psát SQL (ani jiný QL)
- Masivní podpora cachování
  - Primární, sekundární cache
- Podpora pro synchronizaci databázového schématu a doménového modelu
  - hbm2ddl



# ... Hibernate vs. JDBC (pracnost)



Zdroj: Michal Pravda, Diplomová práce na MFF UK, 2007

# Nevýhody

- Přechod na Hibernate „bolí“
  - Vyžaduje jiný způsob myšlení
    - Když chci smazat záznam, musím jej nejdříve načíst
  - Naprosto nefunguje „intuitivní“ přístup
- Plochá learning curve
  - Minimálně 2 dny na základní seznámení
- Omezení designu datového či doménového modelu
  - Aby bylo použití ekonomické





# Základní principy

## Vše se točí okolo Session !

- Obaluje připojení k databázi
- Primární cache – obsahuje všechny načítané a ukládané objekty
- Metody pro CRUD
- Zpravidla přístup „Session per transaction“

# Dotazovací jazyk - HQL

- Podobá se SQL
- Pracuje s třídami místo s tabulkami
- Polymorfní dotazy
  - FROM java.lang.Object
    - Vrátí všechna data z databáze
- Dobrým zvykem je používat Named Queries:
  - Pojmenovaný HQL dotaz
  - HQL je na „jednom“ místě
    - Buď u entity tříd, nebo v zadaném balíčku
  - Rychlejší – Hibernate provede parsing jen jednou





# Dotazovací jazyk - Criteria API

- Alternativa k HQL – dotaz se skládá z Java objektů (žádné HQL, SQL, OQL, ...)
  - Efektivnější
  - IDE nám může napovídat
- Typické použití – vyhledávání podle kritérií
  - Klasické řešení - poskládat z vyhledávacích kritérií SQL (HQL) String.
    - Velmi náchylné k chybám (kompilátor nám nepomůže)
    - Svádí k „trikům“ typu WHERE 1=1



# Konfigurace a metadata

- Konfigurace:
  - XML
    - hibernate.cfg.xml
  - Java
  - Spring
- Mapování mezi doménovým modelem a DB:
  - XML
    - Starší, lepší podpora
    - Nižší produktivita
  - Anotace
    - Standard JPA

# Lazy vs. Eager fetching

- Není rozumné vždy načítat celý objektový graf
  - Možnost inicializovat asociace až při prvním přístupu =>
- Lazy Fetching
  - Implementováno pomocí:
    - Instrumentace byte codu
    - dynamických proxy.
  - Pozor na správnou implementaci metod `equals` a `hashCode`.
  - `LazyInitializationException`







# Když je Hibernatae přiliš horlivý



select souhrnnýli0\_id as id9\_62\_, souhrnnýli0\_datum\_pocátku\_platnosti as datum2\_9\_62\_, souhrnnýli0\_datum\_storna as datum3\_9\_62\_, souhrnnýli0\_id\_limitu as id4\_9\_62\_, souhrnnýli0\_nazev as nazev9\_62\_, souhrnnýli0\_smlouva as smlouva9\_62\_, nazevlimit1\_klic as klic23\_0\_, nazevlimit1\_hodnota as hodnota23\_0\_, pojisteni2\_souhrnný\_limit as souhrnný8\_64\_, pojisteni2\_id as id64\_, pojisteni2\_id as id4\_1\_, pojisteni2\_cislo\_dodatku\_storna as cislo2\_4\_1\_, pojisteni2\_datum\_storna as datum3\_4\_1\_, pojisteni2\_platnost\_od as platnost4\_4\_1\_, pojisteni2\_id\_pojisteni as id5\_4\_1\_, pojisteni2\_max\_pojistne\_plneni as max6\_4\_1\_, pojisteni2\_nazev\_id as nazev14\_4\_1\_, pojisteni2\_predmet\_id as predmet11\_4\_1\_, pojisteni2\_sazebnik\_id as sazebnik13\_4\_1\_, pojisteni2\_sleva as sleva4\_1\_, pojisteni2\_smlouva\_id as smlouva10\_4\_1\_, pojisteni2\_souhrnný\_limit as souhrnný8\_4\_1\_, pojisteni2\_spoluucast\_id as spoluucast12\_4\_1\_, pojisteni2\_typ as typ4\_1\_, nazevpojisi3\_klic as klic25\_2\_, nazevpojisi3\_hodnota as hodnota25\_2\_, policka4\_pojisteni\_id as pojisteni1\_65\_, policko5\_id as policko2\_65\_, policko5\_id as id6\_3\_, policko5\_ciselná\_hodnota as ciselná2\_6\_3\_, policko5\_policko as policko6\_3\_, policko5\_textova\_hodnota as textova3\_6\_3\_, policko5\_vyctova\_hodnota as vyctova4\_6\_3\_, cpolicko6\_id as id48\_4\_, cpolicko6\_typ as typ48\_4\_, hodnoty7\_policko as policko66\_, hodnoty7\_klic as klic66\_, hodnoty7\_klic as klic43\_5\_, hodnoty7\_hodnota as hodnota43\_5\_, hodnoty7\_policko as policko43\_5\_, chodnotapo8\_klic as klic43\_6\_, chodnotapo8\_hodnota as hodnota43\_6\_, chodnotapo8\_policko as policko43\_6\_, predmet9\_id as id7\_7\_, predmet9\_cislo\_dodatku\_storna as cislo2\_7\_7\_, predmet9\_datum\_storna as datum3\_7\_7\_, predmet9\_platnost\_od as platnost4\_7\_7\_, predmet9\_predmet\_id as predmet5\_7\_7\_, predmet9\_misto\_id as misto9\_7\_7\_, predmet9\_nazev\_predmetu as nazev11\_7\_7\_, predmet9\_sazebnik as sazebnik7\_7\_, predmet9\_smlouva\_id as smlouva7\_7\_7\_, predmet9\_specifikace\_predmetu as specifick6\_7\_7\_, predmet9\_typ\_pojistne\_hodnoty as typ10\_7\_7\_, predmet9\_typ\_predmetu as typ8\_7\_7\_, predmet9\_vlastnictvi\_predmetu as vlastni12\_7\_7\_, mistopojis10\_id as id1\_8\_, mistopojis10\_cislo\_dodatku\_storna as cislo2\_1\_8\_, mistopojis10\_datum\_storna as datum3\_1\_8\_, mistopojis10\_platnost\_od as platnost4\_1\_8\_, mistopojis10\_misto\_id as misto5\_1\_8\_, mistopojis10\_cinnost\_id as cinnost9\_1\_8\_, mistopojis10\_popis as popis1\_8\_, mistopojis10\_zona\_id as zona7\_1\_8\_, mistopojis10\_smlouva\_id as smlouva8\_1\_8\_, adresy11\_misto\_pojisteni as misto6\_67\_, adresy11\_id as id67\_, adresy11\_id as id0\_9\_, adresy11\_psc as psc0\_9\_, adresy11\_ulice as ulice0\_9\_, adresy11\_adresa\_id as adresa5\_0\_9\_, adresy11\_misto\_pojisteni as misto6\_0\_9\_, podnikatel12\_klic as klic21\_10\_, podnikatel12\_nazev\_cinnosti as nazev2\_21\_10\_, podnikatel12\_odvetvi as odvetvi21\_10\_, podnikatel13\_klic as klic22\_11\_, podnikatel13\_nazev\_odvetvi as nazev2\_22\_11\_, rizikovapo14\_klic as klic26\_12\_, rizikovapo14\_hodnota as hodnota26\_12\_, rizikovapo14\_platnost\_do as platnost3\_26\_12\_, rizikovapo14\_platnost\_od as platnost4\_26\_12\_, pojistnasm15\_id as id5\_13\_, pojistnasm15\_cetnost\_placeni as cetnost14\_5\_13\_, pojistnasm15\_cislo\_dodatku as cislo2\_5\_13\_, pojistnasm15\_cislo\_navrhu as cislo3\_5\_13\_, pojistnasm15\_cislo\_ps as cislo4\_5\_13\_, pojistnasm15\_konec\_platnosti as konec5\_13\_, pojistnasm15\_pocatek\_platnosti as pocatek6\_5\_13\_, pojistnasm15\_datum\_uzavreni as datum7\_5\_13\_, pojistnasm15\_korespondence as korespo18\_5\_13\_, pojistnasm15\_pobocka\_produkce as pobocka16\_5\_13\_, pojistnasm15\_popis as popis5\_13\_, pojistnasm15\_sleva as sleva5\_13\_, pojistnasm15\_stav as stav5\_13\_, pojistnasm15\_typ\_pojistneho as typ17\_5\_13\_, pojistnasm15\_cislo\_uctu\_pojistnika as cislo10\_5\_13\_, pojistnasm15\_kod\_banky as kod15\_5\_13\_, pojistnasm15\_predcisli\_uctu\_pojistnika as predcisli11\_5\_13\_, pojistnasm15\_specificky\_symbol\_uctu\_pojistnika as specificky12\_5\_13\_, pojistnasm15\_vysledna\_pml as vysledna13\_5\_13\_, cetnostpla16\_klic as klic15\_14\_, cetnostpla16\_hodnota as hodnota15\_14\_, korespondce17\_klic as klic20\_15\_, korespondce17\_hodnota as hodnota20\_15\_, osoby18\_smlouva\_id as smlouva11\_68\_, osoby18\_id as id68\_, osoby18\_id as id3\_16\_, osoby18\_adresa\_id as adresa10\_3\_16\_, osoby18\_koresp\_adresa\_id as koresp14\_3\_16\_, osoby18\_cislo\_pasu as cislo2\_3\_16\_, osoby18\_evidencni\_vypis as evidencni3\_3\_16\_, osoby18\_funkce as funkce3\_16\_, osoby18\_ico as ico3\_16\_, osoby18\_jmeno as jmeno3\_16\_, osoby18\_nazev\_firmy as nazev7\_3\_16\_, osoby18\_prijmeni as prijmeni3\_16\_, osoby18\_rodne\_cislo as rodne9\_3\_16\_, osoby18\_role as role3\_16\_, osoby18\_smlouva\_id as smlouva11\_3\_16\_, osoby18\_statni\_prislusnost as statni15\_3\_16\_, osoby18\_titul\_id as titul13\_3\_16\_, osoby18\_typ as typ3\_16\_, adresa19\_id as id0\_17\_, adresa19\_psc as psc0\_17\_, adresa19\_ulice as ulice0\_17\_, adresa19\_adresa\_id as adresa5\_0\_17\_, adresa19\_misto\_pojisteni as misto6\_0\_17\_, adresa19\_typ as typ0\_17\_, adresa20\_id as id0\_18\_, adresa20\_psc as psc0\_18\_, adresa20\_ulice as ulice0\_18\_, adresa20\_adresa\_id as adresa5\_0\_18\_, adresa20\_misto\_pojisteni as misto6\_0\_18\_, adresa20\_typ as typ0\_18\_, roleosoby21\_klic as klic27\_19\_, roleosoby21\_hodnota as hodnota27\_19\_, statnipris22\_klic as klic29\_20\_, statnipris22\_hodnota as hodnota29\_20\_, titul23\_klic as klic33\_22\_, titul23\_hodnota as hodnota33\_22\_, typosoby24\_klic as klic33\_22\_, typosoby24\_hodnota as hodnota33\_22\_, cislopoboc25\_klic as klic16\_23\_, cislopoboc25\_hodnota as hodnota16\_23\_, pojisteni26\_smlouva\_id as smlouva10\_69\_, pojisteni26\_id as id69\_, pojisteni26\_id as id4\_24\_, pojisteni26\_cislo\_dodatku\_storna as cislo2\_4\_24\_, pojisteni26\_datum\_storna as datum3\_4\_24\_, pojisteni26\_platnost\_od as platnost4\_4\_24\_, pojisteni26\_id\_pojisteni as id5\_4\_24\_, pojisteni26\_nazev\_pojistne\_plneni as max6\_4\_24\_, pojisteni26\_nazev\_id as nazev14\_4\_24\_, pojisteni26\_predmet\_id as predmet11\_4\_24\_, pojisteni26\_sazebnik\_id as sazebnik13\_4\_24\_, pojisteni26\_sleva as sleva4\_24\_, pojisteni26\_smlouva\_id as smlouva10\_4\_24\_, pojisteni26\_souhrnný\_limit as souhrnný8\_4\_24\_, pojisteni26\_spoluucast\_id as spoluucast12\_4\_24\_, pojisteni26\_typ as typ4\_24\_, csazebnik27\_id as id50\_25\_, csazebnik27\_nazev as nazev50\_25\_, csazebnik27\_platnost\_do as platnost3\_50\_25\_, csazebnik27\_platnost\_od as platnost4\_50\_25\_, predmetyp28\_sazebnik as sazebnik70\_, predmetyp28\_platnost\_do as platnost2\_70\_, predmetyp28\_platnost\_od as platnost3\_70\_, predmetyp28\_pojisteni as pojisteni70\_, predmetyp28\_predmet as predmet70\_, cpojisteni29\_id as id47\_26\_, cpojisteni29\_nazev as nazev47\_26\_, konverze30\_pojisteni\_id as pojisteni5\_71\_, konverze30\_id as id71\_, konverze30\_id as id44\_27\_, konverze30\_platnost\_od as platnost3\_44\_27\_, konverze30\_platnost\_od as platnost4\_44\_27\_, konverze30\_pojisteni\_id as pojisteni5\_44\_27\_, okamzikyaa31\_pojisteni\_id as pojisteni1\_72\_, okamzikyaa31\_akce\_id as akce2\_72\_, okamzikyaa31\_okamzik\_id as okamzik3\_72\_, okamzikyaa31\_poradi as poradí72\_, okamzikyaa31\_platnost\_od as platnost5\_72\_, okamzikyaa31\_platnost\_od as platnost6\_72\_, cake32\_id as id41\_28\_, cake32\_java\_program as java2\_41\_28\_, okamzik33\_id as id45\_29\_, okamzik33\_nazev as nazev45\_29\_, policka34\_pojisteni\_id as pojisteni1\_73\_, policka34\_platnost\_do as platnost2\_73\_, policka34\_platnost\_od as platnost3\_73\_, policka34\_policko\_id as policko4\_73\_, cpolicko35\_id as id48\_30\_, cpolicko35\_typ as typ48\_30\_, souhrnnel36\_pojisteni\_id as pojisteni1\_74\_, csouhrnnýl37\_id as souhrne2\_74\_, csouhrnnýl37\_id as id51\_31\_, csouhrnnýl37\_nazev as nazev51\_31\_, cpredmet38\_id as id49\_32\_, cpredmet38\_misto\_poj\_povinne as misto2\_49\_32\_, cpredmet38\_nazev as nazev49\_32\_, cpredmet38\_typ as typ49\_32\_, typpredmet39\_klic as klic37\_33\_, typpredmet39\_hodnota as hodnota37\_33\_, spoluucast40\_klic as klic28\_34\_, spoluucast40\_hodnota as hodnota28\_34\_, ctyppojist41\_klic as klic34\_35\_, ctyppojist41\_hodnota as hodnota34\_35\_, otazky42\_pojisteni\_id as pojisteni1\_75\_, otazky42\_otazka\_id as otazka2\_75\_, otazky42\_platnost\_od as platnost3\_75\_, otazka43\_id as id46\_36\_, otazka43\_poradi as poradí46\_36\_, otazka43\_typ as typ46\_36\_, zavisina44\_pojisteni\_id as pojisteni1\_76\_, zavisina44\_master\_id as master2\_76\_, zavisina44\_pozadovany\_vysledek as pozadovany3\_76\_, otazka45\_id as id46\_37\_, otazka45\_poradi as poradí46\_37\_, otazka45\_typ as typ46\_37\_, predmetry46\_smlouva\_id as smlouva7\_77\_, predmetry46\_id as id77\_, predmetry46\_id as id7\_38\_, predmetry46\_cislo\_dodatku\_storna as cislo2\_7\_38\_, predmetry46\_datum\_storna as datum3\_7\_38\_, predmetry46\_platnost\_od as platnost4\_7\_38\_, predmetry46\_predmet\_id as predmet5\_7\_38\_, predmetry46\_misto\_id as misto9\_7\_38\_, predmetry46\_nazev\_predmetu as nazev11\_7\_38\_, predmetry46\_sazebnik as sazebnik7\_38\_, predmetry46\_smlouva\_id as smlouva7\_7\_38\_, predmetry46\_specifikace\_predmetu as specifick6\_7\_38\_, predmetry46\_typ\_pojistne\_hodnoty as typ10\_7\_38\_, predmetry46\_typ\_predmetu as typ8\_7\_38\_, predmetry46\_vlastnictvi\_predmetu as vlastni12\_7\_38\_, nazevpredm47\_klic as klic24\_39\_, nazevpredm47\_hodnota as hodnota24\_39\_, nemovitost48\_predmet as predmet78\_, nemovitost48\_id as id78\_, nemovitost48\_id as id2\_40\_, nemovitost48\_cena as cena2\_40\_, nemovitost48\_index as index2\_40\_, nemovitost48\_predmet as predmet2\_40\_, nemovitost48\_psc as psc2\_40\_, nemovitost48\_ulice as ulice2\_40\_, nemovitost48\_specifikace as specifick7\_2\_40\_, policka49\_predmet\_id as predmet1\_79\_, policko50\_id as policko2\_79\_, policko50\_id as id6\_41\_, policko50\_ciselná\_hodnota as ciselná2\_6\_41\_, policko50\_textova\_hodnota as textova3\_6\_41\_, policko50\_vyctova\_hodnota as vyctova4\_6\_41\_, csazebnik51\_id as id50\_42\_, csazebnik51\_nazev as nazev50\_42\_, csazebnik51\_platnost\_do as platnost3\_50\_42\_, csazebnik51\_platnost\_od as platnost4\_50\_42\_, typpojistn52\_klic as klic36\_43\_, typpojistn52\_hodnota as hodnota36\_43\_, typpredmet53\_klic as klic37\_44\_, typpredmet53\_hodnota as hodnota37\_44\_, vlastnictv54\_klic as klic39\_45\_, vlastnictv54\_hodnota as hodnota39\_45\_, vozidla55\_predmet as predmet80\_, vozidla55\_vozidlo\_id as vozidlo1\_80\_, vozidla55\_vozidlo\_id as vozidlo1\_12\_46\_, vozidla55\_cena as cena12\_46\_, vozidla55\_index as index12\_46\_, vozidla55\_predmet as predmet12\_46\_, vozidla55\_cena\_nova as cena4\_12\_46\_, vozidla55\_cislo\_karoserie as cislo5\_12\_46\_, vozidla55\_druh as druh12\_46\_, vozidla55\_registraci\_znacka as registra16\_12\_46\_, vozidla55\_rok\_vyroby as rok7\_12\_46\_, vozidla55\_typ\_provedeni as typ9\_12\_46\_, vozidla55\_znacka as znacka12\_46\_, druhovozid156\_klic as klic18\_47\_, druhovozid156\_hodnota as hodnota18\_47\_, typprovede57\_klic as klic38\_48\_, typprovede57\_hodnota as hodnota38\_48\_, znackavozis58\_klic as klic40\_49\_, znackavozis58\_hodnota as hodnota40\_49\_, zarizeni59\_predmet as predmet81\_, zarizeni59\_zarizeni\_id as zarizeni1\_81\_, zarizeni59\_zarizeni\_id as zarizeni1\_13\_50\_, zarizeni59\_cena as cena13\_50\_, zarizeni59\_index as index13\_50\_, zarizeni59\_predmet as predmet13\_50\_, zarizeni59\_rok\_vyroby as rok4\_13\_50\_, zarizeni59\_specifikace as specifick5\_13\_50\_, zarizeni59\_typ as typ13\_50\_, zarizeni59\_vyrobní\_cislo as vyrobní7\_13\_50\_, prilohy60\_smlouva\_id as smlouva4\_82\_, prilohy60\_id as id82\_, prilohy60\_id as id8\_51\_, prilohy60\_druh as druh8\_51\_, prilohy60\_id prilohy as id2\_8\_51\_, prilohy60\_pocet\_stran as pocet3\_8\_51\_, prilohy60\_smlouva\_id as smlouva4\_8\_51\_, druhrpriloh61\_klic as klic17\_52\_, druhrpriloh61\_hodnota as hodnota17\_52\_, souhrnnel62\_smlouva as smlouva83\_, souhrnnel62\_id as id83\_, souhrnnel62\_id as id9\_53\_, souhrnnel62\_datum\_pocátku\_platnosti as datum2\_9\_53\_, souhrnnel62\_datum\_storna as datum3\_9\_53\_, souhrnnel62\_id\_limitu as id4\_9\_53\_, souhrnnel62\_nazev as nazev9\_53\_, souhrnnel62\_smlouva as smlouva9\_53\_, specialniu63\_smlouva\_id as smlouva3\_84\_, specialniu63\_id as id84\_, specialniu63\_cislo\_smlouvy\_id as id10\_54\_, specialniu63\_smlouva\_id as smlouva3\_84\_, specialniu63\_text as text10\_54\_, spravci64\_smlouva\_id as smlouva1\_85\_, spravce65\_id as spravce2\_85\_, spravce65\_id as id11\_55\_, spravce65\_cislo\_pobocky as cislo8\_11\_55\_, spravce65\_cislo\_spravce as cislo2\_11\_55\_, spravce65\_email as email11\_55\_, spravce65\_jmeno as jmeno11\_55\_, spravce65\_osobni\_cislo as osobni5\_11\_55\_, spravce65\_prijmeni as prijmeni11\_55\_, spravce65\_telefon11\_55\_, cislopoboc66\_klic as klic16\_56\_, cislopoboc66\_hodnota as hodnota16\_56\_, stavsmlouv67\_klic as klic30\_57\_, stavsmlouv67\_hodnota as hodnota30\_57\_, typpojistn68\_klic as klic35\_58\_, typpojistn68\_hodnota as hodnota35\_58\_, kodbanky69\_klic as klic19\_59\_, kodbanky69\_kod\_banky as kod2\_19\_59\_, kodbanky69\_nazev\_banky as nazev3\_19\_59\_, ziskatele70\_smlouva\_id as smlouva10\_86\_, ziskatele70\_id as id86\_, ziskatele70\_id as id14\_60\_, ziskatele70\_cislo\_pobocky as cislo9\_14\_60\_, ziskatele70\_cislo\_ziskatele as cislo2\_14\_60\_, ziskatele70\_jmeno as jmeno14\_60\_, ziskatele70\_nazev as nazev14\_60\_, ziskatele70\_osobni\_cislo as osobni5\_14\_60\_, ziskatele70\_podil\_produkce as podil6\_14\_60\_, ziskatele70\_podil\_provize as podil7\_14\_60\_, ziskatele70\_prijmeni as prijmeni14\_60\_, ziskatele70\_smlouva\_id as smlouva10\_14\_60\_, cislopoboc71\_klic as klic16\_61\_, cislopoboc71\_hodnota as hodnota16\_61\_, from\_d\_souhrnný\_limit\_ppr souhrnnýli0\_left outer join c\_nazev\_limitu\_ppr nazevlimit1\_on souhrnnýli0\_nazev=nazevlimit1\_klic left outer join d\_pojisteni\_ppr pojisteni2\_on souhrnnýli0\_id=pojisteni2\_souhrnný\_limit left outer join c\_nazev\_pojisteni\_ppr nazevpojisi3\_on pojisteni2\_nazev\_id=nazevpojisi3\_klic left outer join d\_pojisteni\_policko\_ppr policka4\_on pojisteni2\_id=policka4\_pojisteni\_id left outer join d\_policko\_ppr policko5\_on policka4\_policko\_id=policko5\_id left outer join c\_policko\_ppr policko6\_on policko5\_policko=cpolicko6\_id left outer join c\_policko\_hodnota\_ppr hodnoty7\_on policko6\_id=hodnoty7\_policko left outer join c\_policko\_hodnota\_ppr chodnotapo8\_on policko5\_vyctova\_hodnota=chodnotapo8\_klic left outer join d\_predmet\_ppr predmet9\_on pojisteni2\_predmet\_id=predmet9\_id left outer join d\_misto\_pojisteni\_ppr mistopojis10\_on predmet9\_misto\_id=mistopojis10\_id left outer join d\_adresa\_ppr adresy11\_on mistopojis10\_id=adresy11\_misto\_pojisteni left outer join c\_podnikatelska\_cinnost\_ppr podnikatel12\_on mistopojis10\_cinnost\_id=podnikatel12\_klic left outer join c\_podnikatelske\_odvetvi\_ppr podnikatel13\_on podnikatel12\_odvetvi=podnikatel13\_klic left outer join c\_rizikova\_povodnova\_zona\_ppr rizikovapo14\_on mistopojis10\_zona\_id=rizikovapo14\_klic left outer join d\_pojistna\_smlouva\_ppr pojistnasm15\_on mistopojis10\_smlouva\_id=pojistnasm15\_id left outer join c\_cetnost\_placeni\_ppr cetnostpla16\_on pojistnasm15\_cetnost\_placeni=cetnostpla16\_klic left outer join c\_korespondence\_ppr koresponde17\_on pojistnasm15\_korespondence=koresponde17\_klic left outer join d\_osoba\_ppr osoby18\_on pojistnasm15\_id=osoby18\_smlouva\_id left outer join d\_adresa\_ppr adresa19\_on osoby18\_adresa\_id=adresa19\_id left outer join d\_adresa\_ppr adresa20\_on osoby18\_koresp\_adresa\_id=adresa20\_id left outer join c\_role\_osoby\_ppr roleosoby21\_on osoby18\_role=roleosoby21\_klic left outer join c\_statni\_prislusnost\_ppr statnipris22\_on osoby18\_statni\_prislusnost=statnipris22\_klic left outer join c\_titul\_ppr titul23\_on osoby18\_titul\_id=titul23\_klic left outer join c\_typ\_osoby\_ppr typosoby24\_on osoby18\_typ=typosoby24\_klic left outer join c\_cislo\_pobocky\_ppr cislopoboc25\_on pojistnasm15\_pobocka\_produkce=cislopoboc25\_klic left outer join d\_pojisteni\_ppr pojisteni26\_on pojistnasm15\_id=pojisteni26\_smlouva\_id left outer join c\_sazebnik\_ppr csazebnik27\_on pojisteni26\_sazebnik\_id=csazebnik27\_id left outer join c\_sazebnik\_predmet\_pojisteni\_ppr predmetyp28\_on csazebnik27\_id=predmetyp28\_sazebnik left outer join c\_pojisteni\_ppr pojisteni29\_on predmetyp28\_pojisteni=cpojisteni29\_id left outer join c\_konverze\_ppr konverze30\_on cpojisteni29\_id=konverze30\_pojisteni\_id left outer join c\_pojisteni\_okamzik\_akce\_ppr okamzikyaa31\_on cpojisteni29\_id=okamzikyaa31\_pojisteni\_id left outer join c\_akce\_ppr cake32\_on okamzikyaa31\_akce\_id=cake32\_id left outer join c\_okamzik\_ppr okamzik33\_on okamzikyaa31\_okamzik\_id=okamzik33\_id left outer join c\_pojisteni\_policko\_ppr policka34\_on cpojisteni29\_id=policka34\_pojisteni\_id left outer join c\_policko\_ppr policko35\_on policka34\_policko\_id=cpolicko35\_id left outer join c\_pojisteni\_souhrnnýlimit\_ppr souhrnnel36\_on cpojisteni29\_id=souhrnnel36\_pojisteni\_id left outer join c\_souhrnný\_limit\_ppr csouhrnnýl37\_on souhrnnel36\_souhrnnelLimity\_id=csouhrnnýl37\_id left outer join c\_predmet\_ppr cpredmet38\_on predmetyp28\_predmet=cpredmet38\_id left outer join c\_typ\_predmetu\_ppr typpredmet39\_on cpredmet38\_typ=typpredmet39\_klic left outer join c\_spoluucast\_ppr spoluucast40\_on pojisteni26\_spoluucast\_id=spoluucast40\_klic left outer join c\_typ\_pojisteni\_ppr ctyppojist41\_on pojisteni26\_typ=ctyppojist41\_klic left outer join c\_typpojisteni\_otazka\_ppr otazky42\_on ctyppojist41\_klic=otazky42\_pojisteni\_id left outer join c\_otazka\_ppr cotazka43\_on otazky42\_otazka\_id=cotazka43\_id left outer join c\_otazka\_zavislost\_ppr zavisina44\_on cotazka43\_id=zavisina44\_pojisteni\_id left outer join c\_otazka\_ppr cotazka45\_on zavisina44\_master\_id=cotazka45\_id left outer join d\_predmet\_ppr predmetry46\_on pojistnasm15\_id=predmetry46\_smlouva\_id left outer join c\_nazev\_predmetu\_ppr nazevpredm47\_on predmetry46\_nazev\_nazev\_predmetu=nazevpredm47\_klic left outer join d\_nemovitost\_ppr nemovitost48\_on predmetry46\_id=nemovitost48\_predmet left outer join d\_predmet\_policko\_ppr policka49\_on predmetry46\_id=policka49\_predmet\_id left outer join d\_policko\_ppr policko50\_on policka49\_policko\_id=policko50\_id left outer join c\_sazebnik\_ppr csazebnik51\_on predmetry46\_sazebnik=csazebnik51\_id left outer join c\_typ\_pojistne\_hodnoty\_ppr typpojistn52\_on predmetry46\_typ\_pojistne\_hodnoty=typpojistn52\_klic left outer join c\_typ\_predmetu\_ppr typpredmet53\_on predmetry46\_typ\_predmetu=typpredmet53\_klic left outer join c\_vlastnictvi\_predmetu\_ppr vlastnictv54\_on predmetry46\_vlastnictvi\_predmetu=vlastnictv54\_klic left outer join d\_vozidlo\_ppr vozidla55\_on predmetry46\_id=vozidla55\_predmet left outer join c\_druh\_vozidla\_ppr druhovozid156\_klic left outer join c\_typ\_provedeni\_vozidla\_ppr typprovede57\_on vozidla55\_typ\_provedeni=typprovede57\_klic left outer join c\_znacka\_znackavozis58\_on vozidla55\_znacka=znackavozis58\_klic left outer join d\_zarizeni\_ppr zarizeni59\_on predmetry46\_id=zarizeni59\_predmet left outer join d\_priloha\_ppr prilohy60\_on pojistnasm15\_id=prilohy60\_smlouva\_id left outer join c\_druh\_prilohy60\_ppr druhrpriloh61\_on prilohy60\_druh=druhrpriloh61\_klic left outer join d\_souhrnný\_limit\_ppr souhrnnel62\_on pojistnasm15\_id=souhrnnel62\_smlouva\_id left outer join d\_specialni\_ujednani\_ppr specialniu63\_on pojistnasm15\_id=specialniu63\_smlouva\_id left outer join d\_smlouva\_spravce\_ppr spravci64\_on pojistnasm15\_id=spravci64\_smlouva\_id left outer join d\_spravce\_ppr spravce65\_on spravci64\_spravce\_id=spravce65\_id left outer join c\_cislo\_pobocky\_ppr cislopoboc66\_on spravce65\_cislo\_pobocky=cislopoboc66\_klic left outer join c\_stav\_smlouvy\_ppr stavsmlouv67\_on pojistnasm15\_stav=stavsmlouv67\_klic left outer join c\_typ\_pojistneho\_ppr typpojistn68\_on pojistnasm15\_typ\_pojistneho=typpojistn68\_klic left outer join c\_kod\_banky\_ppr kodbanky69\_on pojistnasm15\_kod\_banky=kodbanky69\_klic left outer join d\_ziskatel\_ppr ziskatele70\_on pojistnasm15\_id=ziskatele70\_smlouva\_id left outer join c\_cislo\_pobocky\_ppr cislopoboc71\_on ziskatele70\_cislo\_pobocky=cislopoboc71\_klic where souhrnnýli0\_id=?

# Pár slov o metodě equals

- Při použití lazy fetchingu pozor na implementaci equals, hashCode a compareTo
  - Hibernate vytváří proxy
- Používat instanceof namísto equals na Class
  - `this.getClass().equals(o.getClass())` je špatně!
  - Zaplatíme ztrátou symetrie (potomek – předeek).
- Používat gettery namísto přímého přístupu k atributům
  - Proxy mají tyto atributy null
- Při použití generovaných primárních klíčů používat v equals business klíč



# Příklad použití

```
...{ // nebezpečí v transakci => Session  
    per operation
```

```
Zakaznik z =  
    session.get(Zakaznik.class, 1);
```

```
z.setJmeno("Daneslav");  
session.update(z);  
...}
```



# Příklad mapování

**@Entity**

**@Table**(name = „objednavka“)

public class **Objednavka** implements Serializable {

**@GeneratedValue**(strategy = TABLE)

**@Id** private Long id;

**@OneToMany**(mappedBy = „objednavka“)

private Set<**Polozka**> zbozi;

// gettery a settery, equals a hashCode

}



# Reálný příklad

```
@FilterDefs({
    @FilterDef(name="CPojisteni_policka", parameters = @ParamDef(name = "platnost", type = "date")),
    @FilterDef(name="CPojisteni_souhrnneLimity", parameters = @ParamDef(name = "platnost", type = "date")),
    @FilterDef(name="CPojisteni_okamzikyAAkce", parameters = @ParamDef(name = "platnost", type = "date")),
    @FilterDef(name="CPojisteni_algoritmy", parameters = @ParamDef(name = "platnost", type = "date"))
})
@Cache(usage = CacheConcurrencyStrategy.READ_ONLY)
@Entity
@Table(name = "c_pojisteni")
public class CPojisteni extends Konfigurace {

    @OneToMany(fetch = FetchType.LAZY,
              mappedBy = "pojisteni")
    @Fetch(value = FetchMode.SELECT)
    @Sort(type = SortType.NATURAL)
    @Filter(name = "CPojisteni_policka",
            condition = TemporalniVztah.FILTER_PLATNOST)
    @Cache(usage = CacheConcurrencyStrategy.READ_ONLY)
    private SortedSet<CPojisteniPolicko> policka;

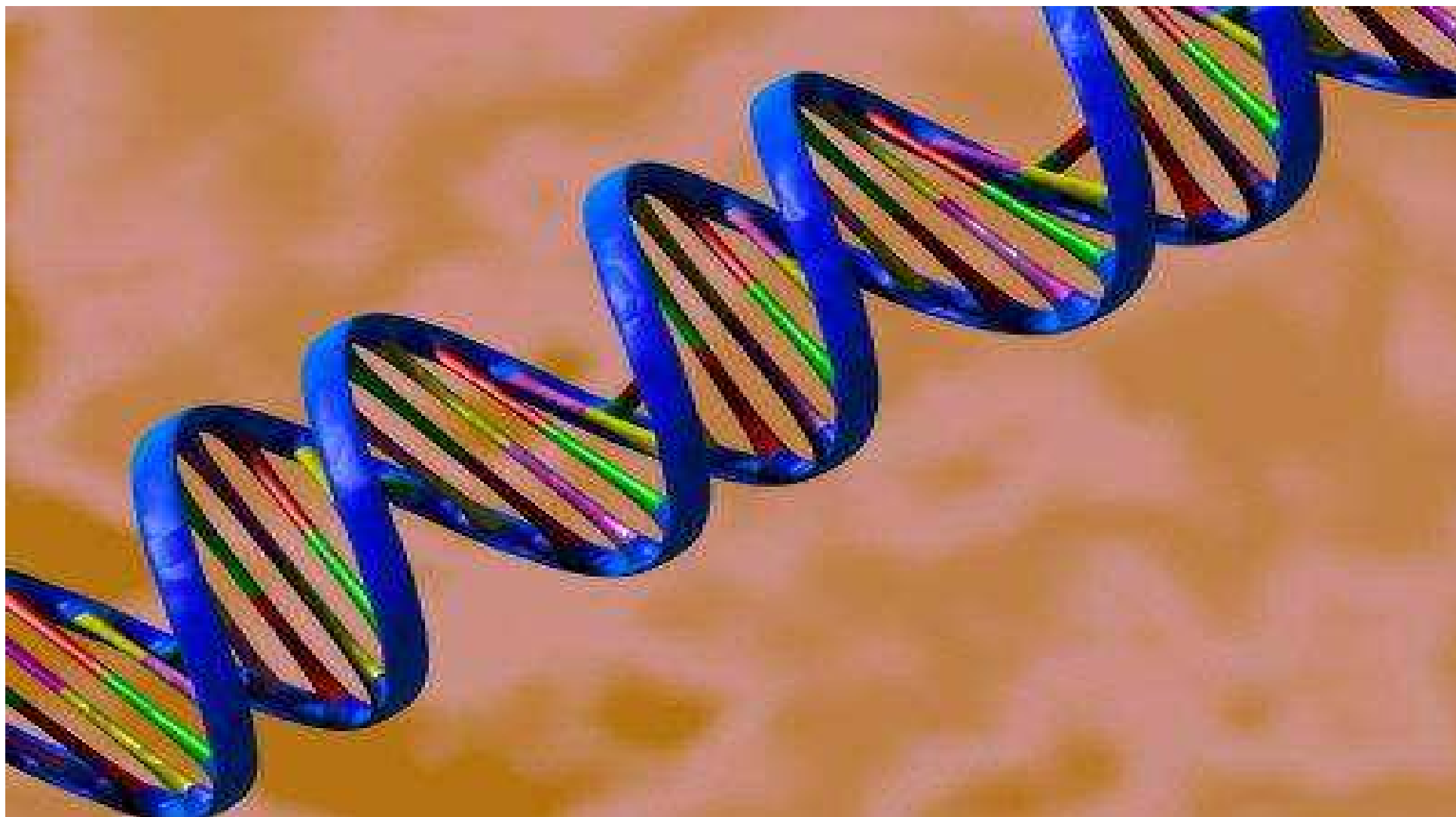
    ...
}
```







# Dědičnost





# Mapování dědičnosti

- Tři strategie:
  - Table per Class (subclass)
  - Single Table per Class Hierarchy
  - Joined Subclass
- Falešná dědičnost - @MappedSuperclass
  - Zdědění atributů a mapování – nevytváří se persistentní hierarchie.
  - Lze použít kdekoliv v hierarchii



# Table per Class

- Pro každého potomka samostatná tabulka
- Implementována pomocí UNIONů
- Potíže s referenční integritou
  - Asociace na předka nelze omezit cizím klíčem
  - Příklad: Osoba – FyzickáOsoba, PrávníckáOsoba (dvě tabulky – fyz\_os a prav\_os).





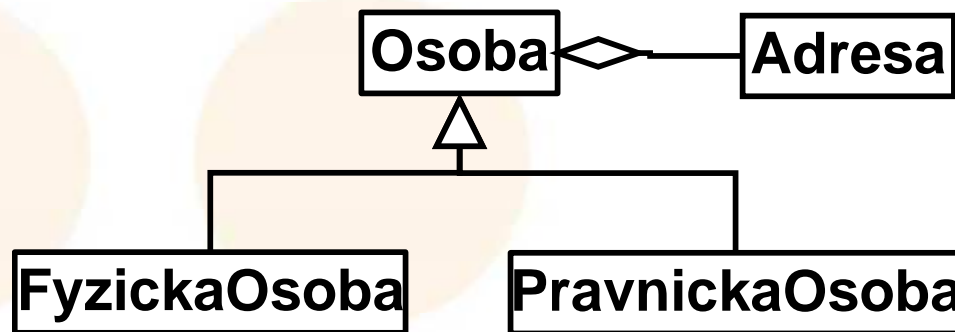
# Single Table per Class Hierarchy

- Všechny třídy v jedné tabulce
  - Nevýhoda - spousta NULL hodnot
- Jak poznat, o který typ se jedná:
  - Default: nový sloupec (discriminator column)
  - SQL Case vracející hodnotu virtuálního discriminator sloupce
- Nejpoužívanější způsob reprezentace dědičnosti



# Joined Subclass

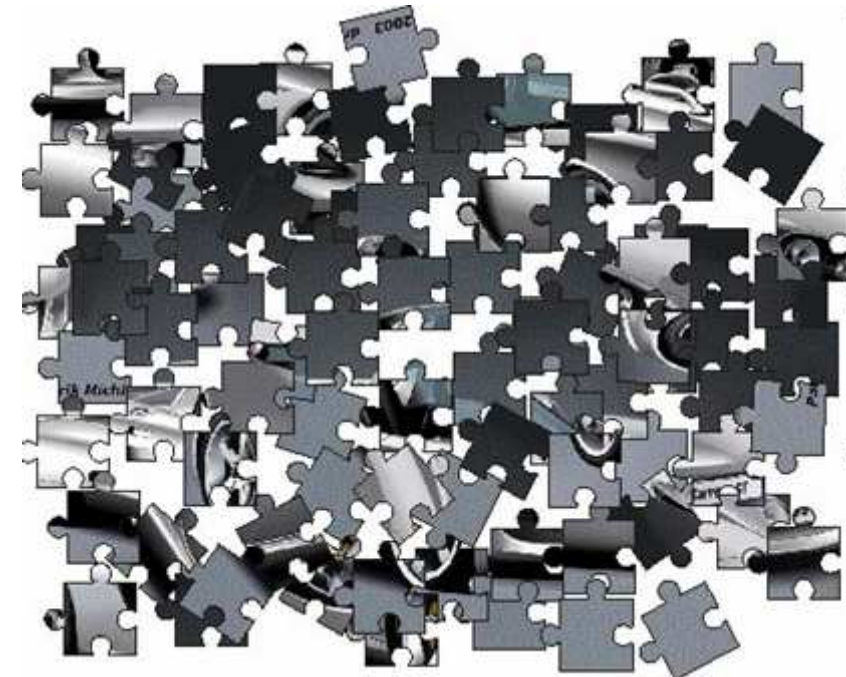
- Tabulka pro předka + pro každého potomka
- Výhodné, pokud je třeba často získat instance předka
- Potomci mohou používat stejný primární klíč jako předek nebo definovat vlastní.
- Nevýhoda – hodně joinů





# Integrace

- Spring
  - Velmi dobrá podpora – HibernateTemplate
- EJB
  - Velmi dobrá podpora – JPA
- Přímá integrace s některými webovými frameworky
  - JSF
  - Tapestry
  - Dokonce i GWT!





# Další zajímavé funkce

- Sekundární cache
  - Existují různé implementace (např. EHCache)
  - Široce konfigurovatelné
- Filtry
  - Umožňují přidat vlastní SQL podmínky při inicializaci kolekcí či seznamu entit.
    - Možno používat parametry
  - Nutno explicitně zapnout
- Custom SQL
  - Pro jednotlivé CRUD operace
- Podpora triggerů, uložených procedur, serializace do XML ...





# Testování

- Základem integrační testy
  - Ověření mapování
  - Kontrola správnosti dotazů
  - Vyšší vrstvy – ověření, že nedochází k `LazyInitializationException`
  - `GenericDaoTest` – ověření funkčnosti základních metod
- Unit testy - pro ORM prakticky nemají smysl
  - Mockování DAO vrstvy – kvůli rychlosti



# Testování

- Frameworky:
  - TestNG, JUnit – téměř nezbytnost
  - DbUnit – uvedení databáze do předem známého stavu
  - EasyMock – „mockování“ (simulace chování)
- Unitils – integrace výše uvedených do jednoho celku



# Kdy použít Hibernate

- Máme **pod kontrolou doménový nebo datový model**
  - Ideálně oba
- **Uložené procedury** se používají jen **v malé míře** nebo **vůbec**
- Používáme podporovanou databázi
  - Podporu lze dopsat (Hibernate dialect)
- Máme **dostatek lidí** se znalostí Hibernate
  - Hibernate je složitý (řeší složitý problém)
  - Požadovaná hloubka znalostí – dle velikosti projektu



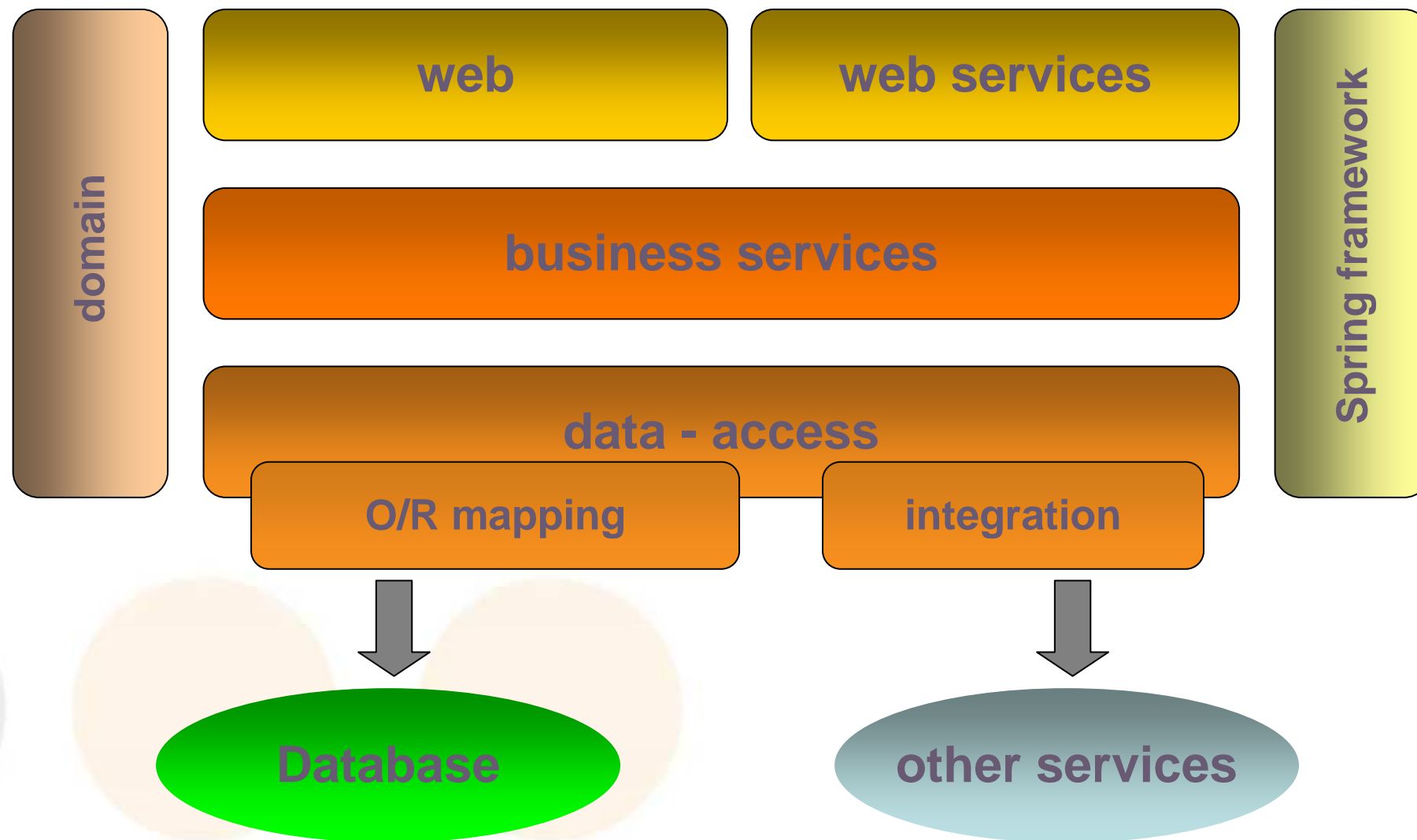


# Hibernate a web



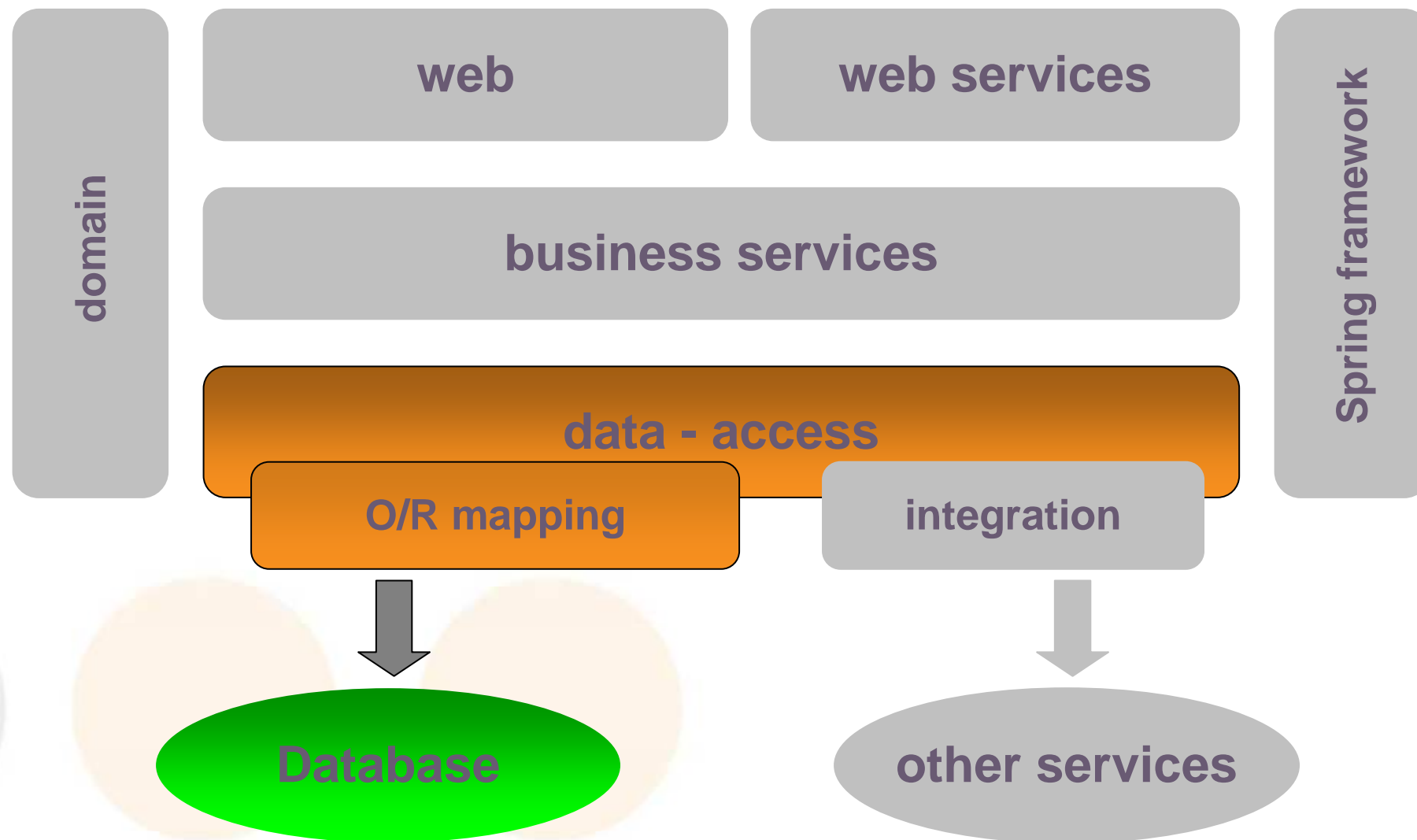


# Architektura webové aplikace





# Persistentní vrstva





# Persistentní vrstva (I.)

- Zdroj dat
  - JDBC connection
  - DataSource (Spring bean, JNDI)
- SessionFactory (Spring bean, JNDI)
- Transakce
  - JDBC API
  - TransactionManager



# Persistentní vrstva (II.)

```
<!-- DataSource -->
<bean id="DS" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
    <property name="driverClassName" value="org.postgresql.Driver" />
    <property name="url" value="jdbc:postgresql://lx2/postgres-utf8" />
    <property name="username" value="postgres" />
    <property name="password" value="postgres" />
</bean>

<!-- Hibernate -->
<bean id="hibernateSessionFactory" class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">
    <property name="dataSource" ref="DS" />
    <property name="configurationClass" value="org.hibernate.cfg.AnnotationConfiguration"/>
    <property name="configLocation" value="classpath:/cz/profinit/kivs/dataprovider/dao/hibernate/hibernate.cfg.xml"/>
</bean>

<bean id="hibernateTxManager" class="org.springframework.orm.hibernate3.HibernateTransactionManager">
    <property name="sessionFactory" ref="hibernateSessionFactory" />
</bean>

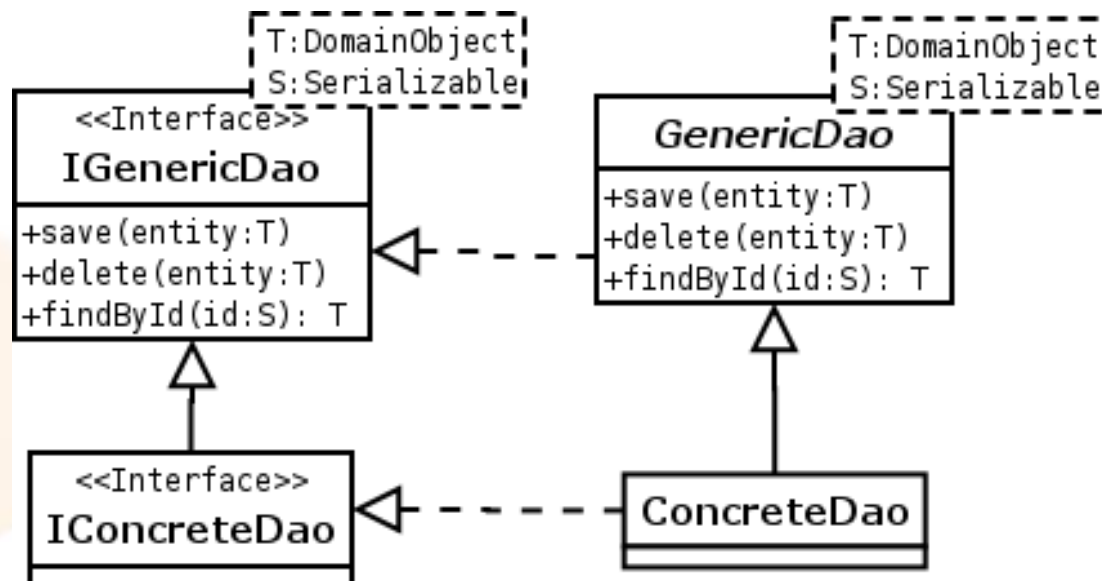
<!-- Annotation TX -->
<tx:annotation-driven transaction-manager="hibernateTxManager" />

<bean id="hibernateTemplate" class="org.springframework.orm.hibernate3.HibernateTemplate">
    <property name="sessionFactory" ref="hibernateSessionFactory" />
</bean>

<!-- DAO -->
<bean id="daoTarget" class="cz.profininit.kivs.dataprovider.dao.hibernate.HibDAO">
    <property name="hibernateTemplate" ref="hibernateTemplate" />
</bean>
```

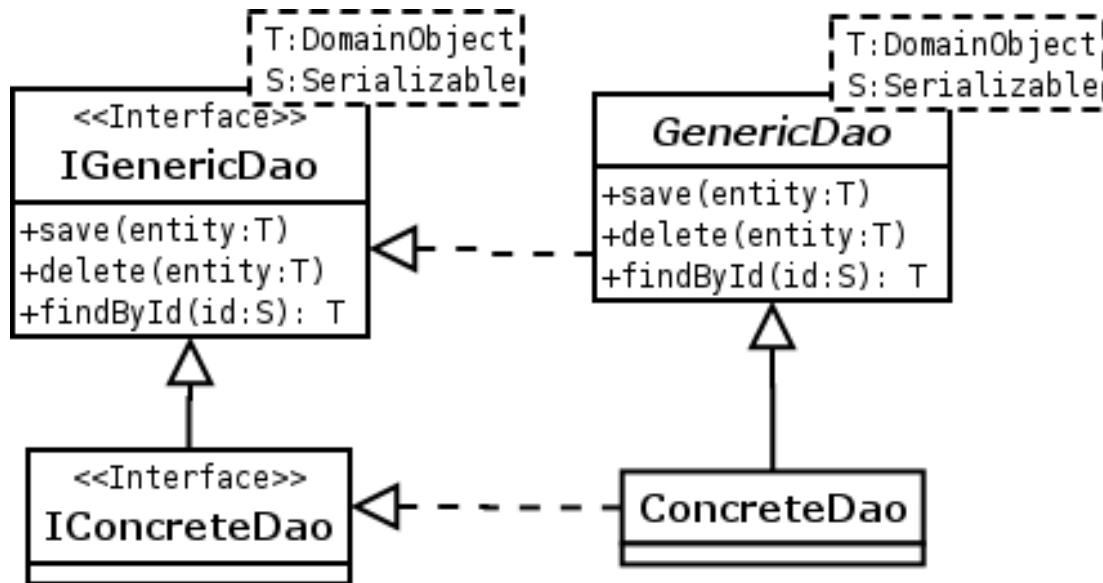
# Design perzistentní vrstvy

- Návrhový vzor GenericDAO
  - Oddělení kódu pracujícího s databází (prostřednictvím ORM) do samostatné vrstvy
  - Pro každý doménový objekt jeden interface a jeho implementace





# GenericDAO



- **IGenericDao**

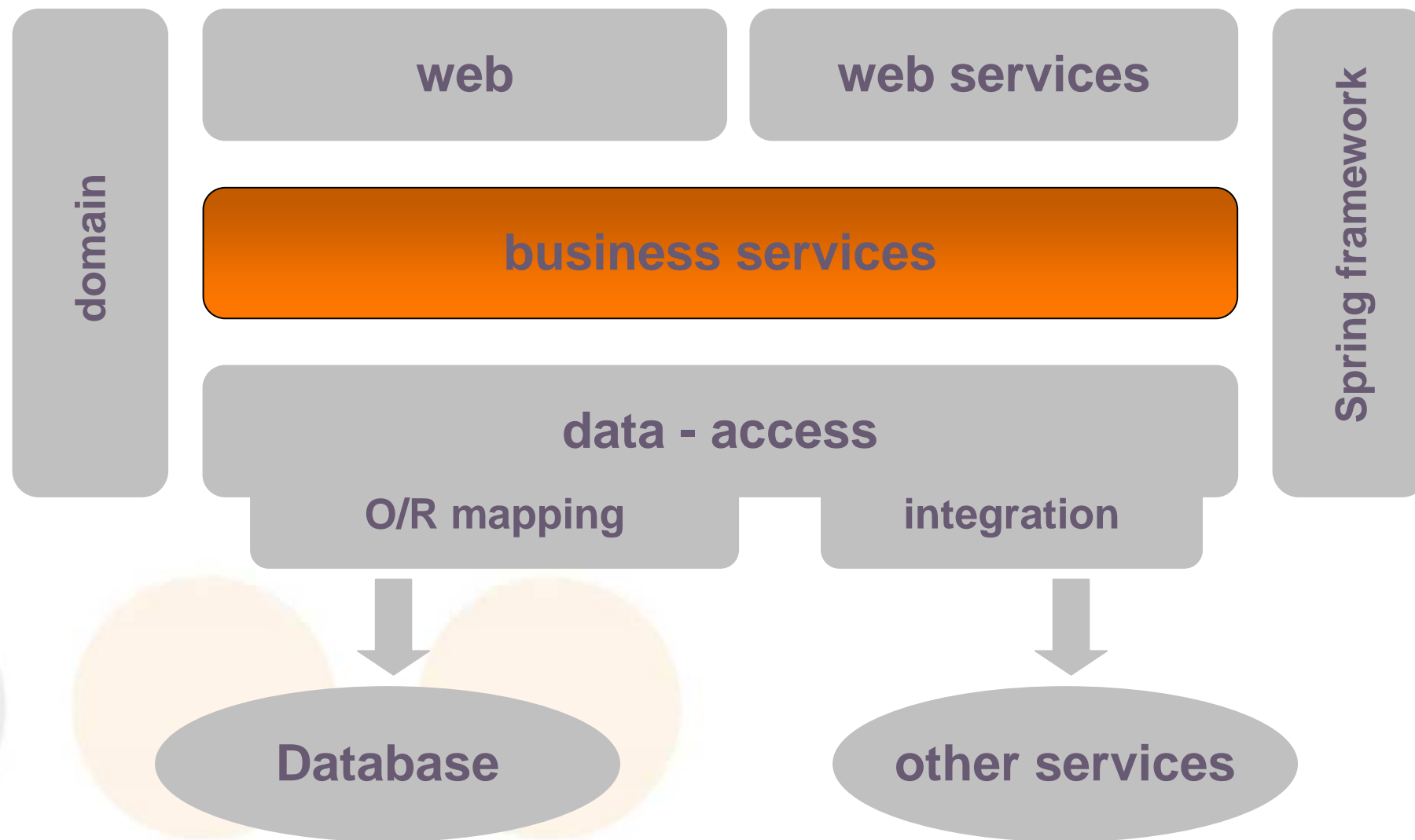
- Metody společné pro všechny DAO
- Typicky obsahuje CRUD operace

- **GenericDAO**

- Implementace základních metod
- Ideální pokud máme k dispozici generické typy (Java 1.5)



# Business vrstva





# Business vrstva (I.)

- Volání DAO



- Řízení transakcí

- Programově (BMT) x deklarativně (CMT)



# Business vrstva (II.)

- Příklad volání persistentní vrstvy

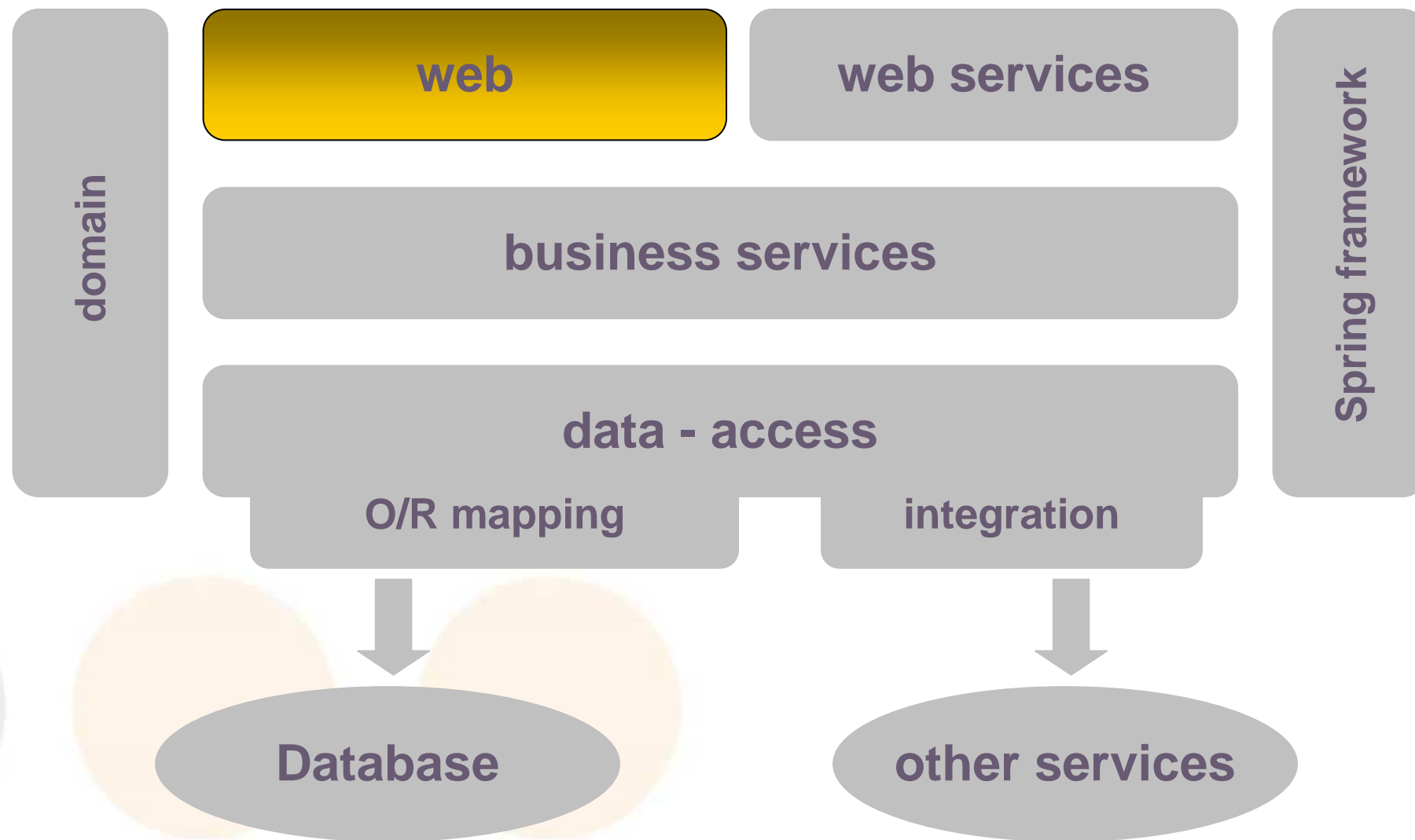
```
@Transactional(propagation = Propagation.REQUIRED, rollbackFor = { ServiceException.class })
public class SluzbaServiceImpl extends AbstractStateManagerAwareService implements SluzbaService {

    public DSluzba getSluzba(DSluzbaId idSluzba) {
        Validate.notNull(idSluzba);

        return (DSluzba) getDao().getById(DSluzba.class, idSluzba);
    }
}
```



# Prezentační vrstva





# Prezentační vrstva (I.)

## Volání business vrstvy

- Pracujeme s objekty doménového modelu







# Jdeme na věc!

```
public class DetailSluzby extends DetailSluzbyBasePage {  
  
    @Inject  
    @Service("sluzbaService")  
    private SluzbaService sluzbaService;  
  
    protected int getSluzbaPocetParametru(DSluzbaId idSluzba) {  
        DSluzba sluzba = sluzbaService.getSluzba(idSluzba);  
        int pocetParametru = sluzba.getDSluzbaParamSet().size();  
        return pocetParametru;  
    }  
}
```

**LazyInitializationException - no session or session was closed!**



# Má ještě něco v záloze?

```
throw new NonUniqueObjectException("a different object with the same identifier value was already associated with the session");
```

```
throw new LazyInitializationException("could not initialize proxy - no Session");
```

```
throw new LazyInitializationException("session is disconnected");
```

```
throw new LazyInitializationException( "entity with lazy properties is not associated with a session" );
```

```
throw new LazyInitializationException("could not initialize proxy - the owning Session is disconnected");
```

```
throw new LazyInitializationException( "session is not connected" );
```

```
throw new LazyInitializationException("could not initialize proxy - the owning Session was closed");
```

```
throw new LazyInitializationException("no session or session was closed");
```

...

# Vyplývá z toho něco?

## Na platnosti Session záleží !



- Eager místo Lazy



- Inicializovat potřebné asociace
- Prodloužit dobu platnosti Session
  - OpenSessionInView pattern



# Shrnutí

- Web je ideální oblastí nasazení
  - Integrace do aplikace není složitá
  - Příliš nezasahuje do návrhu aplikace
- 
- Nutné dodržovat zásady „správného chování“



# Hibernate v desktop aplikacích





# Hibernate v desktop aplikacích

- Hibernate navržen pro web
  - Request – response
    - Request zpracováván v rámci jednoho vlákna
- Desktop aplikace
  - Zpravidla řízené událostmi
  - Práce ve více vláknech
- Základní problémy
  - Předpoklad krátké životnosti Session
  - Session není thread safe





# Základní přístupy

- Detached objects (viz. dále)
- Stále otevřená Session
  - Problém s primární cache
  - Není thread safe
- Žádný lazy fetching ve view vrstvě
  - Simulace chování webu
  - Pracné



# Referenční aplikace

- Slouží pro sjednávání pojistných smluv
- Rozsah ~ 1000 MD
- Technologie:
  - Eclipse RCP
  - Spring
  - Hibernate
  - Apache Derby (emdedded režim)



# Detached objects

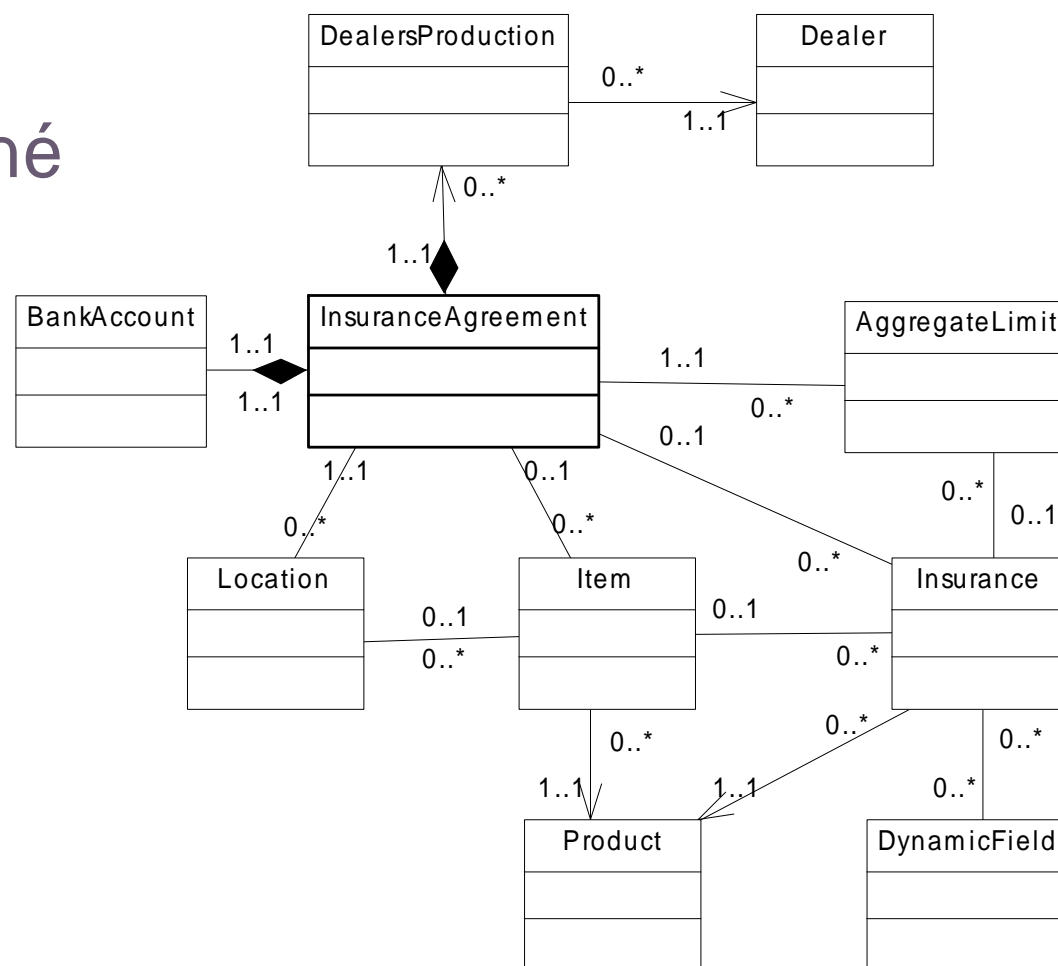
- Načte se celý objektový graf
  - Jedno volání metody
  - V našem případě cca 60 tříd
- Není to pomalé?
  - Načítání a ukládání smlouvy trvá řádově sekundy
  - Pak už je práce s aplikací rychlá





# Zvolené řešení

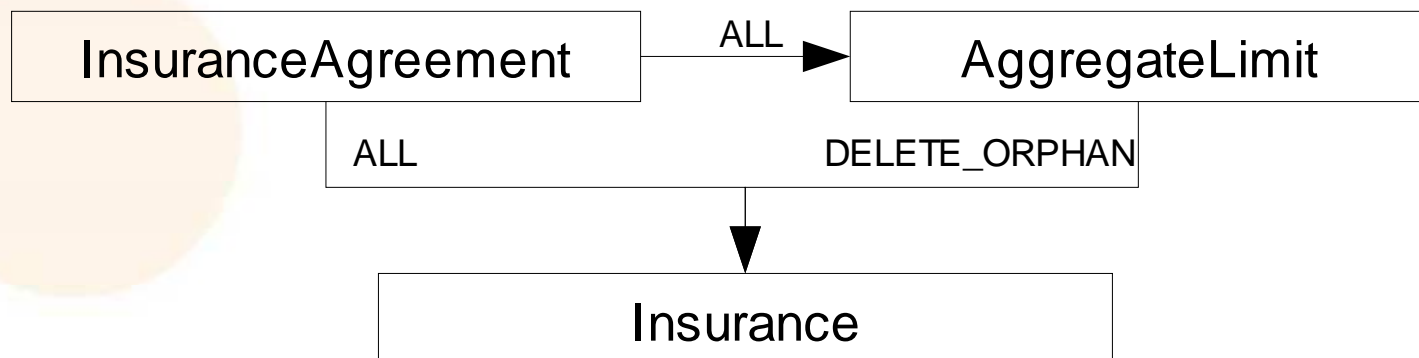
- Jeden kořenový objekt (pojistná smlouva)
  - Jediný, který se načítá a ukládá explicitně
  - Ostatní – kaskády
- Všechny vazby obousměrné
  - Výjimkou číselníky
- Eager fetching





# Problémy

- Správná definice kaskád
  - „Vraždění sirotků“ – all-delete-orphans
- 2-souvislost části objektového grafu
  - K objektu A se lze dostat z objektu B dvěma cestami
  - Vadí kaskádám
- Správné nastavení eager fetchingu
  - Nelze použít joiny!





# Proč nešlo použít joiny?



select souhrnnýli0\_id as id9\_62\_, souhrnnýli0\_datum\_pocátku\_platnosti as datum2\_9\_62\_, souhrnnýli0\_datum\_storna as datum3\_9\_62\_, souhrnnýli0\_id\_limitu as id4\_9\_62\_, souhrnnýli0\_nazev as nazev9\_62\_, souhrnnýli0\_smlouva as smlouva9\_62\_, nazevlimit1\_klic as klic23\_0\_, nazevlimit1\_hodnota as hodnota23\_0\_, pojisteni2\_souhrnný\_limit as souhrnný8\_64\_, pojisteni2\_id as id64\_, pojisteni2\_id as id4\_1\_, pojisteni2\_cislo\_dodatku\_storna as cislo2\_4\_1\_, pojisteni2\_datum\_storna as datum3\_4\_1\_, pojisteni2\_platnost\_od as platnost4\_4\_1\_, pojisteni2\_id\_pojisteni as id5\_4\_1\_, pojisteni2\_max\_pojistne\_plneni as max6\_4\_1\_, pojisteni2\_nazev\_id as nazev14\_4\_1\_, pojisteni2\_predmet\_id as predmet11\_4\_1\_, pojisteni2\_sazebnik\_id as sazebnik13\_4\_1\_, pojisteni2\_sleva as sleva4\_1\_, pojisteni2\_smlouva\_id as smlouva10\_4\_1\_, pojisteni2\_souhrnný\_limit as souhrnný8\_4\_1\_, pojisteni2\_spoluucast\_id as spoluucast12\_4\_1\_, pojisteni2\_typ as typ4\_1\_, nazevpojisi3\_klic as klic25\_2\_, nazevpojisi3\_hodnota as hodnota25\_2\_, policka4\_pojisteni\_id as pojisteni1\_65\_, policko5\_id as policko2\_65\_, policko5\_id as id6\_3\_, policko5\_ciselná\_hodnota as ciselná\_2\_6\_3\_, policko5\_policko as policko6\_3\_, policko5\_textova\_hodnota as textova3\_6\_3\_, policko5\_vyctova\_hodnota as vyctova4\_6\_3\_, cpolicko6\_id as id48\_4\_, cpolicko6\_typ as typ48\_4\_, hodnoty7\_policko as policko66\_, hodnoty7\_klic as klic66\_, hodnoty7\_klic as klic43\_5\_, hodnoty7\_hodnota as hodnota43\_5\_, hodnoty7\_policko as policko43\_5\_, chodnotapo8\_klic as klic43\_6\_, chodnotapo8\_hodnota as hodnota43\_6\_, chodnotapo8\_policko as policko43\_6\_, predmet9\_id as id7\_7\_, predmet9\_cislo\_dodatku\_storna as cislo2\_7\_7\_, predmet9\_datum\_storna as datum3\_7\_7\_, predmet9\_platnost\_od as platnost4\_7\_7\_, predmet9\_predmet\_id as predmet5\_7\_7\_, predmet9\_misto\_id as misto9\_7\_7\_, predmet9\_nazev\_predmetu as nazev11\_7\_7\_, predmet9\_sazebnik as sazebnik7\_7\_, predmet9\_smlouva\_id as smlouva7\_7\_7\_, predmet9\_specifikace\_predmetu as specifick6\_7\_7\_, predmet9\_typ\_pojistne\_hodnoty as typ10\_7\_7\_, predmet9\_typ\_predmetu as typ8\_7\_7\_, predmet9\_vlastnictvi\_predmetu as vlastni12\_7\_7\_, mistopojisi10\_id as id1\_8\_, mistopojisi10\_cislo\_dodatku\_storna as cislo2\_1\_8\_, mistopojisi10\_datum\_storna as datum3\_1\_8\_, mistopojisi10\_platnost\_od as platnost4\_1\_8\_, mistopojisi10\_misto\_id as misto5\_1\_8\_, mistopojisi10\_cinnost\_id as cinnost9\_1\_8\_, mistopojisi10\_popis as popis1\_8\_, mistopojisi10\_zona\_id as zona7\_1\_8\_, mistopojisi10\_smlouva\_id as smlouva8\_1\_8\_, adresy11\_misto\_pojisteni as misto6\_67\_, adresy11\_id as id67\_, adresy11\_id as id0\_9\_, adresy11\_psc as psc0\_9\_, adresy11\_ulice as ulice0\_9\_, adresy11\_adresa\_id as adresa5\_0\_9\_, adresy11\_misto\_pojisteni as misto6\_0\_9\_, podnikatel12\_klic as klic21\_10\_, podnikatel12\_nazev\_cinnosti as nazev2\_21\_10\_, podnikatel12\_odvetvi as odvetvi21\_10\_, podnikatel13\_klic as klic22\_11\_, podnikatel13\_nazev\_odvetvi as nazev2\_22\_11\_, rizikovapo14\_klic as klic26\_12\_, rizikovapo14\_hodnota as hodnota26\_12\_, rizikovapo14\_platnost\_do as platnost3\_26\_12\_, rizikovapo14\_platnost\_od as platnost4\_26\_12\_, pojistnasm15\_id as id5\_13\_, pojistnasm15\_cetnost\_placeni as cetnost14\_5\_13\_, pojistnasm15\_cislo\_dodatku as cislo2\_5\_13\_, pojistnasm15\_cislo\_navrhu as cislo3\_5\_13\_, pojistnasm15\_cislo\_ps as cislo4\_5\_13\_, pojistnasm15\_konec\_platnosti as konec5\_13\_, pojistnasm15\_pocatek\_platnosti as pocatek6\_5\_13\_, pojistnasm15\_datum\_uzavreni as datum7\_5\_13\_, pojistnasm15\_korespondence as korespo18\_5\_13\_, pojistnasm15\_pobocka\_produkce as pobocka16\_5\_13\_, pojistnasm15\_popis as popis5\_13\_, pojistnasm15\_sleva as sleva5\_13\_, pojistnasm15\_stav as stav5\_13\_, pojistnasm15\_typ\_pojistneho as typ17\_5\_13\_, pojistnasm15\_cislo\_uctu\_pojistnika as cislo10\_5\_13\_, pojistnasm15\_kod\_banky as kod15\_5\_13\_, pojistnasm15\_predcisli\_uctu\_pojistnika as predcisli11\_5\_13\_, pojistnasm15\_specificky\_symbol\_uctu\_pojistnika as specificky12\_5\_13\_, pojistnasm15\_vysledna\_pml as vysledna13\_5\_13\_, cetnostplai6\_klic as klic15\_14\_, cetnostplai6\_hodnota as hodnota15\_14\_, koresponde17\_klic as klic20\_15\_, koresponde17\_hodnota as hodnota20\_15\_, osoby18\_smlouva\_id as smlouva11\_68\_, osoby18\_id as id68\_, osoby18\_id as id3\_16\_, osoby18\_adresa\_id as adresa10\_3\_16\_, osoby18\_koresp\_adresa\_id as koresp14\_3\_16\_, osoby18\_cislo\_pasu as cislo2\_3\_16\_, osoby18\_evidencni\_vypis as evidencni3\_3\_16\_, osoby18\_funkce as funkce3\_16\_, osoby18\_ico as ico3\_16\_, osoby18\_jmeno as jmeno3\_16\_, osoby18\_nazev\_firmy as nazev7\_3\_16\_, osoby18\_prijmeni as prijmeni3\_16\_, osoby18\_rodne\_cislo as rodne9\_3\_16\_, osoby18\_role as role3\_16\_, osoby18\_smlouva\_id as smlouva11\_3\_16\_, osoby18\_statni\_prislusnost as statni15\_3\_16\_, osoby18\_titul\_id as titul13\_3\_16\_, osoby18\_typ as typ3\_16\_, adresa19\_id as id0\_17\_, adresa19\_psc as psc0\_17\_, adresa19\_ulice as ulice0\_17\_, adresa19\_adresa\_id as adresa5\_0\_17\_, adresa19\_misto\_pojisteni as misto6\_0\_17\_, adresa19\_typ as typ0\_17\_, adresa20\_id as id0\_18\_, adresa20\_psc as psc0\_18\_, adresa20\_ulice as ulice0\_18\_, adresa20\_adresa\_id as adresa5\_0\_18\_, adresa20\_misto\_pojisteni as misto6\_0\_18\_, adresa20\_typ as typ0\_18\_, roleosoby21\_klic as klic27\_19\_, roleosoby21\_hodnota as hodnota27\_19\_, statnipris22\_klic as klic29\_20\_, statnipris22\_hodnota as hodnota29\_20\_, titul23\_klic as klic33\_22\_, titul23\_hodnota as hodnota33\_22\_, typosoby24\_hodnota as hodnota33\_22\_, cislopoboc25\_klic as klic26\_23\_, cislopoboc25\_hodnota as hodnota16\_23\_, pojisteni26\_smlouva\_id as smlouva10\_69\_, pojisteni26\_id as id69\_, pojisteni26\_id as id4\_24\_, pojisteni26\_cislo\_dodatku\_storna as cislo2\_4\_24\_, pojisteni26\_datum\_storna as datum3\_4\_24\_, pojisteni26\_platnost\_od as platnost4\_4\_24\_, pojisteni26\_id\_pojisteni as id5\_4\_24\_, pojisteni26\_max\_pojistne\_plneni as max6\_4\_24\_, pojisteni26\_nazev\_id as nazev14\_4\_24\_, pojisteni26\_predmet\_id as predmet11\_4\_24\_, pojisteni26\_sazebnik\_id as sazebnik13\_4\_24\_, pojisteni26\_sleva as sleva4\_24\_, pojisteni26\_smlouva\_id as smlouva10\_4\_24\_, pojisteni26\_souhrnný\_limit as souhrnný8\_4\_24\_, pojisteni26\_spoluucast\_id as spoluucast12\_4\_24\_, pojisteni26\_typ as typ4\_24\_, csazebnik27\_id as id0\_25\_, csazebnik27\_nazev as nazev50\_25\_, csazebnik27\_platnost\_do as platnost3\_50\_25\_, csazebnik27\_platnost\_od as platnost4\_50\_25\_, predmetypa28\_sazebnik as sazebnik70\_, predmetypa28\_platnost\_do as platnost2\_70\_, predmetypa28\_platnost\_od as platnost3\_70\_, predmetypa28\_pojisteni as pojisteni70\_, predmetypa28\_predmet as predmet70\_, cpojisteni29\_id as id47\_26\_, cpojisteni29\_nazev as nazev47\_26\_, konverze30\_pojisteni\_id as pojisteni5\_71\_, konverze30\_id as id71\_, konverze30\_id as id44\_27\_, konverze30\_platnost\_od as platnost3\_44\_27\_, konverze30\_platnost\_od as platnost4\_44\_27\_, konverze30\_pojisteni\_id as pojisteni5\_44\_27\_, okamzikyaa31\_pojisteni\_id as pojisteni1\_72\_, okamzikyaa31\_akce\_id as akce2\_72\_, okamzikyaa31\_okamzik\_id as okamzik3\_72\_, okamzikyaa31\_poradi as poradi72\_, okamzikyaa31\_platnost\_od as platnost5\_72\_, okamzikyaa31\_platnost\_od as platnost6\_72\_, cake32\_id as id41\_28\_, cake32\_java\_program as java2\_41\_28\_, cokamzik33\_id as id45\_29\_, cokamzik33\_nazev as nazev45\_29\_, policka34\_platnost\_od as platnost2\_73\_, policka34\_platnost\_od as platnost3\_73\_, policka34\_policko\_id as policka0\_73\_, cpolicko35\_id as id48\_30\_, cpolicko35\_typ as typ48\_30\_, souhrnnel36\_pojisteni\_id as pojisteni1\_74\_, csouhrnnýl37\_id as souhrne2\_74\_, csouhrnnýl37\_id as id51\_31\_, csouhrnnýl37\_nazev as nazev51\_31\_, cpredmet38\_id as id49\_32\_, cpredmet38\_misto\_poj\_povinne as misto2\_49\_32\_, cpredmet38\_nazev as nazev49\_32\_, cpredmet38\_typ as typ49\_32\_, typpredmet39\_klic as klic37\_33\_, typpredmet39\_hodnota as hodnota37\_33\_, spoluucast40\_klic as klic28\_34\_, spoluucast40\_hodnota as hodnota28\_34\_, ctyppojist41\_klic as klic34\_35\_, ctyppojist41\_hodnota as hodnota34\_35\_, otazky42\_pojisteni\_id as pojisteni1\_75\_, otazky42\_otazka\_id as otazka2\_75\_, otazky42\_platnost\_od as platnost3\_75\_, otazky42\_platnost\_od as platnost4\_75\_, otazka43\_id as id46\_36\_, otazka43\_poradi as poradi46\_36\_, otazka43\_typ as typ46\_36\_, zavisina44\_pojisteni\_id as pojisteni1\_76\_, zavisina44\_master\_id as master2\_76\_, zavisina44\_pozadovany\_vysledek as pozadovany3\_76\_, otazka45\_id as id46\_37\_, otazka45\_poradi as poradi46\_37\_, otazka45\_typ as typ46\_37\_, predmetypa46\_smlouva\_id as smlouva7\_77\_, predmetypa46\_id as id77\_, predmetypa46\_id as id7\_38\_, predmetypa46\_cislo\_dodatku\_storna as cislo2\_7\_38\_, predmetypa46\_datum\_storna as datum3\_7\_38\_, predmetypa46\_platnost\_od as platnost4\_7\_38\_, predmetypa46\_predmet\_id as predmet5\_7\_38\_, predmetypa46\_misto\_id as misto9\_7\_38\_, predmetypa46\_nazev\_predmetu as nazev11\_7\_38\_, predmetypa46\_sazebnik as sazebnik7\_38\_, predmetypa46\_smlouva\_id as smlouva7\_7\_38\_, predmetypa46\_specifikace\_predmetu as specifick6\_7\_38\_, predmetypa46\_typ\_pojistne\_hodnoty as typ10\_7\_38\_, predmetypa46\_typ\_predmetu as typ8\_7\_38\_, predmetypa46\_vlastnictvi\_predmetu as vlastni12\_7\_38\_, nazevypredm47\_klic as klic24\_39\_, nazevypredm47\_hodnota as hodnota24\_39\_, nemovitost48\_predmet as predmet78\_, nemovitost48\_id as id78\_, nemovitost48\_id as id2\_40\_, nemovitost48\_cena as cena2\_40\_, nemovitost48\_index as index2\_40\_, nemovitost48\_predmet as predmet2\_40\_, nemovitost48\_psc as psc2\_40\_, nemovitost48\_ulice as ulice2\_40\_, nemovitost48\_specifikace as specifick7\_2\_40\_, policka49\_predmet\_id as predmet1\_79\_, policko50\_id as policko2\_79\_, policko50\_id as id6\_41\_, policko50\_ciselná\_hodnota as ciselná\_2\_6\_41\_, policko50\_textova\_hodnota as textova3\_6\_41\_, policko50\_vyctova\_hodnota as vyctova4\_6\_41\_, csazebnik51\_id as id50\_42\_, csazebnik51\_nazev as nazev50\_42\_, csazebnik51\_platnost\_do as platnost3\_50\_42\_, csazebnik51\_platnost\_od as platnost4\_50\_42\_, typpojistn52\_klic as klic36\_43\_, typpojistn52\_hodnota as hodnota36\_43\_, typpredmet53\_klic as klic37\_44\_, typpredmet53\_hodnota as hodnota37\_44\_, vlastnictvi54\_klic as klic39\_45\_, vlastnictvi54\_hodnota as hodnota39\_45\_, vozidla55\_predmet as predmet80\_, vozidla55\_vozidlo\_id as vozidlo1\_80\_, vozidla55\_vozidlo\_id as vozidlo1\_12\_46\_, vozidla55\_cena as cena12\_46\_, vozidla55\_index as index12\_46\_, vozidla55\_predmet as predmet12\_46\_, vozidla55\_cena\_nova as cena4\_12\_46\_, vozidla55\_cislo\_karoserie as cislo5\_12\_46\_, vozidla55\_druh as druh12\_46\_, vozidla55\_registracni\_znacka as registra6\_12\_46\_, vozidla55\_rok\_vyroby as rok7\_12\_46\_, vozidla55\_typ\_provedeni as typ9\_12\_46\_, vozidla55\_znacka as znacka12\_46\_, druhovozidli56\_hodnota as hodnota18\_47\_, typprovede57\_klic as klic38\_48\_, typprovede57\_hodnota as hodnota38\_48\_, znackavozis58\_klic as klic0\_49\_, znackavozis58\_hodnota as hodnota40\_49\_, zarizeni59\_predmet as predmet81\_, zarizeni59\_zarizeni\_id as zarizeni1\_81\_, zarizeni59\_zarizeni\_id as zarizeni1\_13\_50\_, zarizeni59\_cena as cena2\_13\_50\_, zarizeni59\_index as index13\_50\_, zarizeni59\_predmet as predmet13\_50\_, zarizeni59\_rok\_vyroby as rok4\_13\_50\_, zarizeni59\_specifikace as specifick5\_13\_50\_, zarizeni59\_typ as typ13\_50\_, zarizeni59\_vyrobní\_cislo as vyrobní7\_13\_50\_, prilohy60\_smlouva\_id as smlouva4\_82\_, prilohy60\_id as id82\_, prilohy60\_id as id8\_51\_, prilohy60\_druh as druh8\_51\_, prilohy60\_id prilohy as id2\_8\_51\_, prilohy60\_pocet\_stran as pocet3\_8\_51\_, prilohy60\_smlouva\_id as smlouva4\_8\_51\_, druhrpriloh61\_klic as klic17\_52\_, druhrpriloh61\_hodnota as hodnota17\_52\_, souhrnnel62\_smlouva as smlouva83\_, souhrnnel62\_id as id83\_, souhrnnel62\_id as id9\_53\_, souhrnnel62\_datum\_pocátku\_platnosti as datum2\_9\_53\_, souhrnnel62\_datum\_storna as datum3\_9\_53\_, souhrnnel62\_id\_limitu as id4\_9\_53\_, souhrnnel62\_nazev as nazev9\_53\_, souhrnnel62\_smlouva as smlouva9\_53\_, specialniui63\_smlouva\_id as smlouva34\_84\_, specialniui63\_id as id84\_, specialniui63\_smlouva\_id as smlouva3\_10\_54\_, specialniui63\_text as text10\_54\_, spravci64\_smlouva\_id as smlouva1\_85\_, spravce65\_id as spravce2\_85\_, spravce65\_id as id11\_55\_, spravce65\_cislo\_pobocky as cislo8\_11\_55\_, spravce65\_cislo\_spravce as cislo2\_11\_55\_, spravce65\_email as email11\_55\_, spravce65\_jmeno as jmeno11\_55\_, spravce65\_osobni\_cislo as osobni5\_11\_55\_, spravce65\_prijmeni as prijmeni11\_55\_, spravce65\_telefon11\_55\_, cislopoboc66\_klic as klic16\_56\_, cislopoboc66\_hodnota as hodnota16\_56\_, stavsmilouv67\_klic as klic30\_57\_, stavsmilouv67\_hodnota as hodnota30\_57\_, typpojistn68\_klic as klic35\_58\_, typpojistn68\_hodnota as hodnota35\_58\_, kodbanky69\_klic as klic19\_59\_, kodbanky69\_kod\_banky as kod2\_19\_59\_, kodbanky69\_nazev\_banky as nazev3\_19\_59\_, ziskatele70\_smlouva\_id as smlouva10\_86\_, ziskatele70\_id as id86\_, ziskatele70\_id as id14\_60\_, ziskatele70\_cislo\_pobocky as cislo9\_14\_60\_, ziskatele70\_cislo\_ziskatele as cislo2\_14\_60\_, ziskatele70\_jmeno as jmeno14\_60\_, ziskatele70\_nazev as nazev14\_60\_, ziskatele70\_osobni\_cislo as osobni5\_14\_60\_, ziskatele70\_podil\_produkce as podil6\_14\_60\_, ziskatele70\_podil\_provize as podil7\_14\_60\_, ziskatele70\_prijmeni as prijmeni14\_60\_, ziskatele70\_smlouva\_id as smlouva10\_14\_60\_, cislopoboc71\_klic as klic16\_61\_, cislopoboc71\_hodnota as hodnota16\_61\_, frou souhrnný\_limit\_ppr souhrnnýli0\_left outer join c\_nazev\_limitu\_ppr nazevlimit1\_on souhrnnýli0\_nazev=nazevlimit1\_klic left outer join d\_pojisteni\_ppr pojisteni2\_on souhrnnýli0\_id=pojisteni2\_souhrnný\_limit left outer join c\_nazev\_pojisteni\_ppr nazevpojisi3\_on pojisteni2\_nazev\_id=nazevpojisi3\_klic left outer join d\_pojisteni\_policko\_ppr policka4\_on pojisteni2\_id=policka4\_pojisteni\_id left outer join c\_policko\_ppr policko5\_on policka4\_policko\_id=policko5\_id left outer join c\_policko\_ppr policko6\_on policko5\_policko\_id=policko6\_id left outer join c\_policko\_hodnota\_ppr hodnoty7\_on policko6\_id=hodnoty7\_policko left outer join c\_policko\_hodnota\_ppr chodnotapo8\_on policko5\_vyctova\_hodnota=chodnotapo8\_klic left outer join d\_predmet\_ppr predmet9\_on pojisteni2\_predmet\_id=predmet9\_id left outer join d\_misto\_pojisteni\_ppr mistopojisi10\_on predmet9\_misto\_id=mistopojisi10\_id left outer join d\_adresa\_ppr adresy11\_on mistopojisi10\_id=adresy11\_misto\_pojisteni left outer join c\_podnikatelska\_cinnost\_ppr podnikatel12\_on mistopojisi10\_cinnost\_id=podnikatel12\_klic left outer join c\_podnikatelska\_odvetvi\_ppr podnikatel13\_on podnikatel12\_odvetvi=podnikatel13\_klic left outer join c\_rizikova\_povodnova\_zona\_ppr rizikovapo14\_on mistopojisi10\_zona\_id=rizikovapo14\_klic left outer join d\_pojistna\_smlouva\_ppr pojistnasm15\_on mistopojisi10\_smlouva\_id=pojistnasm15\_id left outer join c\_cetnost\_placeni\_ppr cetnostplai6\_on pojistnasm15\_cetnost\_placeni=cetnostplai6\_klic left outer join c\_korespondence\_ppr koresponde17\_on pojistnasm15\_korespondence=koresponde17\_klic left outer join d\_osoba\_ppr osoby18\_on pojistnasm15\_id=osoby18\_smlouva\_id left outer join d\_adresa\_ppr adresa19\_on osoby18\_adresa\_id=adresa19\_id left outer join d\_adresa\_ppr adresa20\_on osoby18\_koresp\_adresa\_id=adresa20\_id left outer join c\_role\_osoby\_ppr roleosoby21\_on osoby18\_role=roleosoby21\_klic left outer join c\_statni\_prislusnost\_ppr statnipris22\_on osoby18\_statni\_prislusnost=statnipris22\_klic left outer join c\_titul\_ppr titul23\_on osoby18\_titul\_id=titul23\_klic left outer join c\_typ\_osoby\_ppr typosoby24\_on osoby18\_typ=typosoby24\_klic left outer join c\_cislo\_pobocky\_ppr cislopoboc25\_on pojistnasm15\_pobocka\_produkce=cislopoboc25\_klic left outer join d\_pojisteni\_ppr pojisteni26\_on pojistnasm15\_id=pojisteni26\_smlouva\_id left outer join c\_sazebnik\_ppr csazebnik27\_on pojisteni26\_sazebnik\_id=csazebnik27\_id left outer join c\_sazebnik\_predmet\_pojisteni\_ppr predmetypa28\_on csazebnik27\_id=predmetypa28\_sazebnik left outer join c\_pojisteni\_ppr cpojisteni29\_on predmetypa28\_pojisteni=cpojisteni29\_id left outer join c\_konverze\_ppr konverze30\_on cpojisteni29\_id=konverze30\_pojisteni\_id left outer join c\_pojisteni\_okamzik\_akce\_ppr okamzikyaa31\_on cpojisteni29\_id=okamzikyaa31\_pojisteni\_id left outer join c\_akce\_ppr cake32\_on okamzikyaa31\_akce\_id=cake32\_id left outer join c\_okamzik\_ppr cokamzik33\_on okamzikyaa31\_okamzik\_id=cokamzik33\_id left outer join c\_pojisteni\_policko\_ppr policka34\_on cpojisteni29\_id=policka34\_pojisteni\_id left outer join c\_policko\_ppr policko35\_on policka34\_policko\_id=policko35\_id left outer join c\_pojisteni\_souhrnnýlimit\_ppr souhrnnel36\_on cpojisteni29\_id=souhrnnel36\_pojisteni\_id left outer join c\_souhrnný\_limit\_ppr csouhrnnýl37\_on souhrnnel36\_souhrnnelLimit\_id=csouhrnnýl37\_id left outer join c\_predmet\_ppr cpredmet38\_on predmetypa28\_predmet=cpredmet38\_id left outer join c\_typ\_predmetu\_ppr typpredmet39\_on cpredmet38\_typ=typpredmet39\_klic left outer join c\_spoluucast\_ppr spoluucast40\_on pojisteni26\_spoluucast\_id=spoluucast40\_klic left outer join c\_typ\_pojisteni\_ppr ctyppojist41\_on pojisteni26\_typ=ctyppojist41\_klic left outer join c\_typpojisteni\_otazka\_ppr otazky42\_on ctyppojist41\_klic=otazky42\_pojisteni\_id left outer join c\_otazka\_ppr cotazka43\_on otazky42\_otazka\_id=cotazka43\_id left outer join c\_otazka\_zavislost\_ppr zavisina44\_on cotazka43\_id=zavisina44\_pojisteni\_id left outer join c\_otazka\_ppr cotazka45\_on zavisina44\_master\_id=cotazka45\_id left outer join d\_predmet\_ppr predmetypa46\_on pojistnasm15\_id=predmetypa46\_smlouva\_id left outer join c\_nazev\_predmetu\_ppr nazevypredm47\_on predmetypa46\_nazev\_predmetu=nazevypredm47\_klic left outer join d\_nemovitost\_ppr nemovitost48\_on predmetypa46\_id=nemovitost48\_predmet left outer join d\_predmet\_policko\_ppr policka49\_on predmetypa46\_id=policka49\_predmet\_id left outer join d\_policko\_ppr policko50\_on policka49\_policko\_id=policko50\_id left outer join c\_sazebnik\_ppr csazebnik51\_on predmetypa46\_sazebnik=csazebnik51\_id left outer join c\_typ\_pojistne\_hodnoty\_ppr typpojistn52\_on predmetypa46\_typ\_pojistne\_hodnoty=typpojistn52\_klic left outer join c\_typ\_predmetu\_ppr typpredmet53\_on predmetypa46\_typ\_predmetu=typpredmet53\_klic left outer join c\_vlastnictvi\_predmetu\_ppr vlastnictvi54\_on predmetypa46\_vlastnictvi\_predmetu=vlastnictvi54\_klic left outer join d\_vozidlo\_ppr vozidla55\_on predmetypa46\_id=vozidla55\_predmet left outer join c\_druh\_vozidla\_ppr druhovozidli56\_klic left outer join c\_typ\_provedeni\_vozidla\_ppr typprovede57\_on vozidla55\_typ\_provedeni=typprovede57\_klic left outer join c\_znacka\_vozidla\_ppr znackavozis58\_on vozidla55\_znacka=znackavozis58\_klic left outer join d\_zarizeni\_ppr zarizeni59\_on predmetypa46\_id=zarizeni59\_predmet left outer join d\_priloha\_ppr prilohy60\_on pojistnasm15\_id=prilohy60\_smlouva\_id left outer join c\_druh\_prilohy60\_ppr druhrpriloh61\_on prilohy60\_druh=druhrpriloh61\_klic left outer join d\_souhrnný\_limit\_ppr souhrnnel62\_on pojistnasm15\_id=souhrnnel62\_smlouva\_id left outer join d\_specialni\_ujednani\_ppr specialniui63\_on pojistnasm15\_id=specialniui63\_smlouva\_id left outer join d\_smlouva\_spravce\_ppr spravci64\_on pojistnasm15\_id=spravci64\_smlouva\_id left outer join d\_spravce\_ppr spravce65\_on spravci64\_spravce\_id=spravce65\_id left outer join c\_cislo\_pobocky\_ppr cislopoboc66\_on spravce65\_cislo\_pobocky=cislopoboc66\_klic left outer join c\_stav\_smlouvy\_ppr stavsmilouv67\_on pojistnasm15\_stav=stavsmilouv67\_klic left outer join c\_typ\_pojistneho\_ppr typpojistn68\_on pojistnasm15\_typ\_pojistneho=typpojistn68\_klic left outer join c\_kod\_banky\_ppr kodbanky69\_on pojistnasm15\_kod\_banky=kodbanky69\_klic left outer join d\_ziskatel\_ppr ziskatele70\_on pojistnasm15\_id=ziskatele70\_smlouva\_id left outer join c\_cislo\_pobocky\_ppr cislopoboc71\_on ziskatele70\_cislo\_pobocky=cislopoboc71\_klic where souhrnnýli0\_id=?

# Výhody zvoleného řešení

- Databázové schéma se generuje z doménového modelu
  - Pro vývojáře je update zpravidla zcela transparentní
- Z 10 členů týmu se o persistenci dat staral 1
  - Konzistence, nízká režie



Vysoká produktivita







# Představení knihovny iBatis

Aneb alternativy tu jsou





# iBatis

- Produkt Apache Foundation, open source
- Knihovna pro ORM
- Finální verze 1 od roku 2002

**iBatis** 





# Projektové prostředí

- Sun App Server
- Spring
- Tapestry
  
- Naše zkušenosti:
  - Bezproblémová integrace
  - Splňuje všechny požadavky





# Obsah

- Proč hledat novou cestu?
- Co je iBatis?
- Praktické ukázky
- Nevýhody
- Diskuse





# Proč ne Hibernate?



● 

```
select souhrnnlyi0_id as id9_62_, souhrnnlyi0_datum_pocátku_platnosti as datum2_9_62_, souhrnnlyi0_datum_storna as datum3_9_62_,  
souhrnnlyi0_id_limitu as id4_9_62_, souhrnnlyi0_nazev as nazev9_62_, souhrnnlyi0_smlouva as smlouva9_62_, nazevlimit1_klic as klic23_0_,  
nazevlimit1_hodnota as hodnota23_0_, pojisteni2_souhrnnny_limit as souhrnnny8_64_, pojisteni2_id as id64_, pojisteni2_id as id4_1_,  
pojisteni2_cislo_dodatku_storna as cislo2_4_1_, pojisteni2_datum_storna as datum3_4_1_, pojisteni2_platnost_od as platnost4_4_1_,  
pojisteni2_id_pojisteni as id5_4_1_, pojisteni2_max_pojistne_plneni as max6_4_1_, pojisteni2_nazev_id as nazev14_4_1_,  
pojisteni2_predmet_id as predmet11_4_1_, pojisteni2_sazebnik_id as sazebnik13_4_1_, pojisteni2_sleva as sleva4_1_, pojisteni2_smlouva_id  
as smlouva10_4_1_, pojisteni2_souhrnnny_limit as souhrnnny8_4_1_, pojisteni2_spoluucast_id as spoluucast12_4_1_, pojisteni2_typ as typ4_1_,  
nazevpojisi3_klic as klic25_2_, nazevpojisi3_hodnota as hodnota25_2_, policka4_pojisteni_id as pojisteni1_65_, policko5_id as policko2_65_,  
policko5_id as id6_3_, policko5_ciselna_hodnota as ciselna2_6_3_, policko5_policko as policko6_3_, policko5_textova_hodnota as  
textova3_6_3_, policko5_vyctova_hodnota as vyctova4_6_3_, cpolicko6_id as id48_4_, cpolicko6_typ as typ48_4_, hodnoty7_policko as  
policko66_, hodnoty7_klic as klic66_, hodnoty7_klic as klic43_5_, hodnoty7_hodnota as hodnota43_5_, hodnoty7_policko as policko43_5_,  
chodnotapo8_klic as klic43_6_, chodnotapo8_hodnota as hodnota43_6_, chodnotapo8_policko as policko43_6_, predmet9_id as id7_7_,  
predmet9_cislo_dodatku_storna as cislo2_7_7_, predmet9_datum_storna as datum3_7_7_, predmet9_platnost_od as platnost4_7_7_,  
predmet9_predmet_id as predmet5_7_7_, predmet9_misto_id as misto9_7_7_...
```

● LazyInitializationException



# Proč ne Hibernate? Shrnutí

- Některé úkony (např. uložené procedury) jsou zbytečně složité.
- „Donutit Hibernate dělat přesně to, co chceme, může být problém.“





# Proč ne JDBC?

```
query = "select * from users";
appendWhere = true;

if (userFilter.getName() != null) {
    if (appendWhere) query += " where ";
    else query += " and ";
    appendWhere = false;
    query += " name like '" +
        userFilter.getName() + "'";
}
```

```
if (userFilter.getSurname() != null) {
    if (appendWhere) query += " where ";
    else query += " and ";
    appendWhere = false;
    query += " surname like '" +
        userFilter.getSurname() + "' ";
}
...
```

- Duplicita kódu
- Mezery, uvozovky atd.
- SQL injection
- ...





# Proč ne JDBC? Pokračování

- Dotazy roztržštěné v kódu
- Častá duplicita celých dotazů
- O mapování result set na objekty nemluvě





# Střední cesta iBatis

- Použití čistého SQL
- O/R mapování na jednom místě (XML)
- Všechny dotazy na jednom místě (XML)
- XML konstrukce pro řešení předpon a přípon volitelných částí dotazu, SQL injection, ...
- Nejlepší přirovnání: Vylepšený Spring JDBC template



# Příklady

- Nastavení mapování
- Zápis SQL dotazu
- Použití předchozího se Spring JDBC template
- Pokročilé: dynamicky vkládané položky dotazu
- Pokročilé: Řešení vazby 1:N





# O/R mapování

- XML zápis pro O/R mapování:

```
<resultMap id="resultMap.DataType" class="cz.profinit.DataType">
```

```
  <result column="DATA_TYPE" jdbcType="DECIMAL" property="dataType"  
    typeHandler="cz.profinit.DataTypeTypeHandler"/>
```

```
  <result column="DESCRIPTION" jdbcType="VARCHAR"  
    property="description" />
```

```
  <result column="ACTIVE" property="active"  
    typeHandler="cz.profinit.YesNoTypeHandler" />
```

```
  ...  
</resultMap>
```

- Mapování sloupce result setu na atribut třídy
  - Lze použít type handlers



# Jednoduchý select

- XML zápis pro jednoduchý select:

```
<select id="DataTypes.select" resultMap="resultMap.DataType">  
    select DATA_TYPE, DESCRIPTION, ACTIVE from DATA_TYPES  
</select>
```



# Použití pomocí Spring template

- Předchozí select dotaz a mapování mohou být použity z kódu takto:

```
public class DataTypesDaoIBatis extends SqlMapClientDaoSupport {  
    public List<DataType> getDataTypes() {  
        ...  
        return  
        getSqlMapClientTemplate().queryForList("DataTypes.select");  
        ...  
    }  
}
```



# Dynamicky vkládané položky

```
<select id="DataTypes.select" resultMap="resultMap.DataType"
  parameterClass="cz.profinit.DataType">
  select DATA_TYPE, DESCRIPTION, ACTIVE from DATA_TYPES
  <dynamic prepend="WHERE">
    <isNotNull prepend="AND" property="dataType">
      DATA_TYPE = #dataType:DECIMAL#
    </isNotNull>
  </dynamic>
</select>
```

```
public class DataTypesDaoIBatis extends SqlMapClientDaoSupport {
  public DataType getDataTypeById(Integer id) {
    ...
    return (DataType)
    getSqlMapClientTemplate().queryForObject("DataTypes.select", new
    DataType(id));
    ...
  }
}
```





# Řešení vazby 1:N

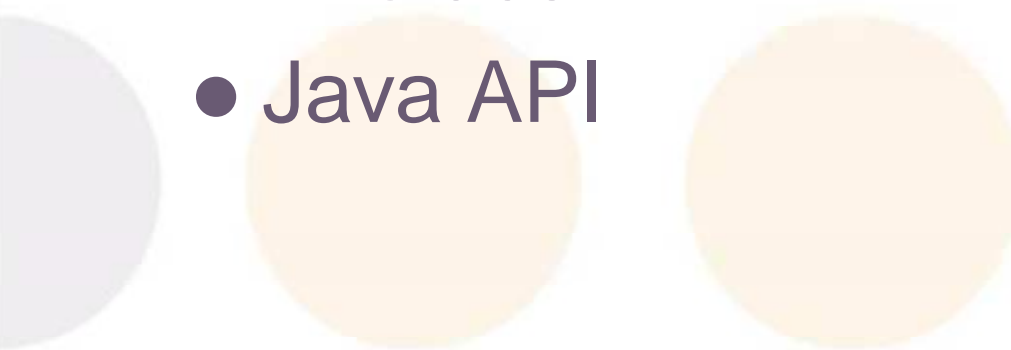
```
<resultMap id="resultMap.FlightDetail"  
  class="cz.profinit.mobilegha.model.FlightDetail">  
  <result property="dataType" resultMap="resultMap.DataTypes"/>  
  ...  
</resultMap>  
  
<select id="FlightDetails.select"  
  resultMap="resultMap.FlightDetail">  
  select  
    FD.IMMATRICULATION, ...  
    DT.DATA_TYPE, ... from  
    FLIGHT_DETAILS FD  
    left join DATA_TYPES DT on FD.DATA_TYPE = DT.DATA_TYPE  
</select>
```



# Další vlastnosti iBatis

- Caching
- Lazy fetching

Ve verzi 3.0:

- Interface binding
  - Anotace
  - Java API
- 



# Nevýhody (naše zkušenosti)

- Někdy nutnost vytváření třídy parametru dotazu pouze kvůli iBatis
- Záludné chyby při nejednoznačném pojmenování sloupců tabulky, zdlouhavé odstraňování takových chyb
- Svádí k programování v XML





# Kdy použít iBatis

- Chcete / potřebujete mít věci pod kontrolou
  - SQL
  - Nejčastější důvod
- Používá se hodně uložených procedur
- Chcete dodržovat dobré praktiky a mít SQL mimo kód
  - A nechcete objevovat kolo (tj. napsat si to sami).



# Závěr





# Závěr

- ORM se hodí téměř na každý projekt
  - Pokud chceme mít věci pod kontrolou – iBatis
  - V ostatních případech Hibernate
- Hlavní výhody:
  - Rychlejší vývoj
  - Lepší udržitelnost (snazší provádění změn)
- Hlavní nevýhody:
  - Někdy neprůhledné chování (Hibernate)
    - Vychází z complexity samotného problému (viz. impedance mismatch)
  - Občas omezuje design datového nebo doménového modelu



# Diskuse

- Komentáře
- Otázky
- Připomínky
- Upřesnění
- Poznámky
- ...

