Inheritance Mapping in Hibernate (Table Per Class)

# Maven Configuration(pom.xml)

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>table-per-class</groupId>

<artifactId>table-per-class</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>many-to-many</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<spring.version>4.1.2.RELEASE</spring.version>

<spring.security.version>3.2.3.RELEASE</spring.security.version>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.0</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.1.30</version>

</dependency>

<!-- C3P0 library -->

<dependency>

<groupId>com.mchange</groupId>

<artifactId>c3p0</artifactId>

<version>0.9.5</version>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>4.3.7.Final</version>

</dependency>

</dependencies>

</project>

# SQL Configuration (ddl.sql)

**drop** **table** if **exists** person2;

**CREATE** **TABLE** person2 (

person\_id BIGINT(10) **NOT** **NULL** AUTO\_INCREMENT,

name **VARCHAR**(50) **NULL** **DEFAULT** **NULL**,

deptName **VARCHAR**(50) **NULL** **DEFAULT** **NULL**,

discriminator **VARCHAR**(20) **NOT** **NULL**,

**PRIMARY** **KEY** (person\_id)

);

**select** \* **from** person2;

## Spring-Hibernate Configuration (app-context.xml)

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:aop=*"http://www.springframework.org/schema/aop"* xmlns:jee=*"http://www.springframework.org/schema/jee"*

xmlns:tx=*"http://www.springframework.org/schema/tx"* xmlns:jdbc=*"http://www.springframework.org/schema/jdbc"*

xmlns:osgi=*"http://www.springframework.org/schema/osgi"* xmlns:security=*"http://www.springframework.org/schema/security"*

xsi:schemaLocation=*"http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-3.0.xsd*

*http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*

*http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.0.xsd*

*http://www.springframework.org/schema/jee http://www.springframework.org/schema/jee/spring-jee-3.0.xsd*

*http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-3.0.xsd*

*http://www.springframework.org/schema/jdbc http://www.springframework.org/schema/jdbc/spring-jdbc-3.0.xsd*

*http://www.springframework.org/schema/osgi http://www.springframework.org/schema/osgi/spring-osgi.xsd*

*http://www.springframework.org/schema/security http://www.springframework.org/schema/security/spring-security-3.0.3.xsd"*>

<tx:annotation-driven transaction-manager=*"discussionTransactionManager"* />

<bean id=*"dataSourceInternal"* class=*"com.mchange.v2.c3p0.ComboPooledDataSource"*

destroy-method=*"close"*>

<property name=*"driverClass"* value=*"com.mysql.jdbc.Driver"* />

<property name=*"jdbcUrl"* value=*"jdbc:mysql://localhost/test"* />

<property name=*"user"* value=*"deba"* />

<property name=*"password"* value=*"deba"* />

<!-- these are C3P0 properties -->

<property name=*"acquireIncrement"* value=*"5"* />

<property name=*"initialPoolSize"* value=*"5"* />

<property name=*"minPoolSize"* value=*"5"* />

<property name=*"maxPoolSize"* value=*"20"* />

<!-- <property name="maxIdleTime" value="${c3p0.maxIdleTime}" />

<property name="checkoutTimeout" value="${c3p0.checkoutTimeout}" />

<property name="preferredTestQuery" value="${c3p0.preferredTestQuery}" />

<property name="idleConnectionTestPeriod" value="${c3p0.idleConnectionTestPeriod}" />

<property name="maxIdleTimeExcessConnections" value="${c3p0.maxIdleTimeExcessConnections}" />

<property name="numHelperThreads" value="${c3p0.numHelperThreads}" />

<property name="unreturnedConnectionTimeout" value="${c3p0.unreturnedConnectionTimeout}" />

<property name="debugUnreturnedConnectionStackTraces" value="${c3p0.debugUnreturnedConnectionStackTraces}" />

<property name="testConnectionOnCheckin" value="${c3p0.testConnectionOnCheckin}" />

<property name="acquireRetryAttempts" value="${c3p0.acquireRetryAttempts}" />

<property name="acquireRetryDelay" value="${c3p0.acquireRetryDelay}" />

<property name="breakAfterAcquireFailure" value="${c3p0.breakAfterAcquireFailure}" /> -->

</bean>

<!-- This is the lazy DataSource proxy that interacts with the target DataSource once a real statement is sent to the database. Users use this DataSource to set up their Hibernate session factory, which in turn forces the Hibernate second-level cache and also everything that interacts with that Hibernate session factory to use it. -->

<bean id=*"dataSource"* class=*"org.springframework.jdbc.datasource.LazyConnectionDataSourceProxy"*>

<property name=*"targetDataSource"*><ref bean=*"dataSourceInternal"* /></property>

</bean>

<!-- <bean id="hibSessionFactory" class="org.springframework.orm.hibernate3.annotation.AnnotationSessionFactoryBean"> -->

<bean id=*"hibSessionFactory"* class=*"org.springframework.orm.hibernate4.LocalSessionFactoryBean"*>

<property name=*"dataSource"* ref=*"dataSource"* />

<!--<property name="hibernateProperties"> <value> hibernate.show\_sql=true

</value> </property> -->

<property name=*"hibernateProperties"*>

<value>

hibernate.id.new\_generator\_mappings=true,

hibernate.show\_sql=true

<!-- hibernate.hbm2ddl.auto=update -->

<!-- hibernate.current\_session\_context\_class=thread -->

</value>

</property>

<property name=*"annotatedClasses"*>

<list>

<value>com.ddlab.rnd.orm.Person</value>

<value>com.ddlab.rnd.orm.Employee</value>

</list>

</property>

</bean>

</beans>

# Java Source Code

## Person.java

**package** com.ddlab.rnd.orm;

**import** javax.persistence.Column;

**import** javax.persistence.DiscriminatorColumn;

**import** javax.persistence.DiscriminatorType;

**import** javax.persistence.DiscriminatorValue;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.Id;

**import** javax.persistence.Inheritance;

**import** javax.persistence.InheritanceType;

**import** javax.persistence.Table;

@Entity

@Table(name = "PERSON2")

@Inheritance(strategy=InheritanceType.***SINGLE\_TABLE***)

@DiscriminatorColumn(

name="discriminator",

discriminatorType=DiscriminatorType.***STRING***

)

@DiscriminatorValue(value="P")

**public** **class** Person {

@Id

@GeneratedValue

@Column(name = "PERSON\_ID")

**private** Long personId;

@Column(name = "NAME")

**private** String name;

**public** Person() {

}

**public** Person(String name) {

**this**.name = name;

}

**public** Long getPersonId() {

**return** personId;

}

**public** **void** setPersonId(Long personId) {

**this**.personId = personId;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

}

## Employee.java

**package** com.ddlab.rnd.orm;

**import** javax.persistence.Column;

**import** javax.persistence.DiscriminatorValue;

**import** javax.persistence.Entity;

**import** javax.persistence.Table;

@Entity

@Table(name="PERSON2")

@DiscriminatorValue("E")

**public** **class** Employee **extends** Person {

@Column(name="deptName")

**private** String deptName;

**public** Employee() {

}

**public** Employee(String name, String deptName) {

**super**(name);

**this**.deptName = deptName;

}

**public** String getDeptName() {

**return** deptName;

}

**public** **void** setDeptName(String deptName) {

**this**.deptName = deptName;

}

}

## App.java (Test Program)

**package** com.ddlab.rnd.orm;

**import** org.hibernate.HibernateException;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**public** **class** App

{

**public** **static** **void** main( String[] args )

{

ApplicationContext context = **new** ClassPathXmlApplicationContext(

"app-context.xml");

SessionFactory sessionFactory = (SessionFactory) context

.getBean("hibSessionFactory");

Session session = sessionFactory.openSession();

Transaction transaction = **null**;

**try** {

transaction = session.beginTransaction();

Person person = **new** Person("Deb");

session.save(person);

Employee employee = **new** Employee("Deb", "Technology");

session.save(employee);

transaction.commit();

}

**catch** (HibernateException e) {

**if** (transaction != **null**)

transaction.rollback();

e.printStackTrace();

} **finally** {

session.close();

}

System.***out***.println("successfully saved into database");

}

}

Inheritance Mapping in Hibernate (Table Per Sub Class)

# Maven Configuration (pom.xml)

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>table-per-subclass</groupId>

<artifactId>table-per-subclass</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>table-per-subclass</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<spring.version>4.1.2.RELEASE</spring.version>

<spring.security.version>3.2.3.RELEASE</spring.security.version>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.0</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.1.30</version>

</dependency>

<!-- C3P0 library -->

<dependency>

<groupId>com.mchange</groupId>

<artifactId>c3p0</artifactId>

<version>0.9.5</version>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>4.3.7.Final</version>

</dependency>

</dependencies>

</project>

# SQL Confifuration (ddl.sql)

**drop** **table** if **exists** owner3;

**drop** **table** if **exists** employee3;

**drop** **table** if **exists** person3;

**CREATE** **TABLE** person3 (

person\_id BIGINT(20) **NOT** **NULL** AUTO\_INCREMENT,

name **VARCHAR**(50) **NOT** **NULL**,

**PRIMARY** **KEY** (person\_id)

);

**CREATE** **TABLE** employee3 (

person\_id BIGINT(10) **NOT** **NULL**,

deptName **VARCHAR**(50) **NULL** **DEFAULT** **NULL**,

**PRIMARY** **KEY** (person\_id),

**FOREIGN** **KEY** (person\_id) **REFERENCES** person3 (person\_id)

);

**CREATE** **TABLE** owner3 (

person\_id BIGINT(20) **NOT** **NULL**,

city **VARCHAR**(50) **NULL** **DEFAULT** **NULL**,

**PRIMARY** **KEY** (person\_id),

**FOREIGN** **KEY** (person\_id) **REFERENCES** person3 (person\_id)

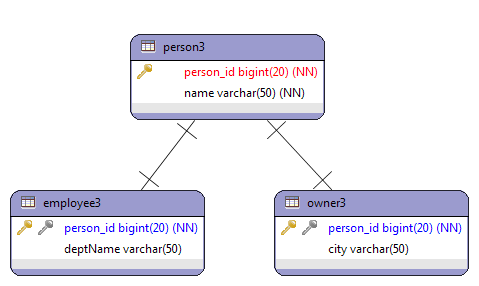
);

**select** \* **from** owner3;

**select** \* **from** employee3;

**select** \* **from** person3;

# Data Model



# Spring-Hibernate Configuration (app-context.xml)

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:aop=*"http://www.springframework.org/schema/aop"* xmlns:jee=*"http://www.springframework.org/schema/jee"*

xmlns:tx=*"http://www.springframework.org/schema/tx"* xmlns:jdbc=*"http://www.springframework.org/schema/jdbc"*

xmlns:osgi=*"http://www.springframework.org/schema/osgi"* xmlns:security=*"http://www.springframework.org/schema/security"*

xsi:schemaLocation=*"http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-3.0.xsd*

*http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*

*http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.0.xsd*

*http://www.springframework.org/schema/jee http://www.springframework.org/schema/jee/spring-jee-3.0.xsd*

*http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-3.0.xsd*

*http://www.springframework.org/schema/jdbc http://www.springframework.org/schema/jdbc/spring-jdbc-3.0.xsd*

*http://www.springframework.org/schema/osgi http://www.springframework.org/schema/osgi/spring-osgi.xsd*

*http://www.springframework.org/schema/security http://www.springframework.org/schema/security/spring-security-3.0.3.xsd"*>

<tx:annotation-driven transaction-manager=*"discussionTransactionManager"* />

<bean id=*"dataSourceInternal"* class=*"com.mchange.v2.c3p0.ComboPooledDataSource"*

destroy-method=*"close"*>

<property name=*"driverClass"* value=*"com.mysql.jdbc.Driver"* />

<property name=*"jdbcUrl"* value=*"jdbc:mysql://localhost/test"* />

<property name=*"user"* value=*"deba"* />

<property name=*"password"* value=*"deba"* />

<!-- these are C3P0 properties -->

<property name=*"acquireIncrement"* value=*"5"* />

<property name=*"initialPoolSize"* value=*"5"* />

<property name=*"minPoolSize"* value=*"5"* />

<property name=*"maxPoolSize"* value=*"20"* />

<!-- <property name="maxIdleTime" value="${c3p0.maxIdleTime}" />

<property name="checkoutTimeout" value="${c3p0.checkoutTimeout}" />

<property name="preferredTestQuery" value="${c3p0.preferredTestQuery}" />

<property name="idleConnectionTestPeriod" value="${c3p0.idleConnectionTestPeriod}" />

<property name="maxIdleTimeExcessConnections" value="${c3p0.maxIdleTimeExcessConnections}" />

<property name="numHelperThreads" value="${c3p0.numHelperThreads}" />

<property name="unreturnedConnectionTimeout" value="${c3p0.unreturnedConnectionTimeout}" />

<property name="debugUnreturnedConnectionStackTraces" value="${c3p0.debugUnreturnedConnectionStackTraces}" />

<property name="testConnectionOnCheckin" value="${c3p0.testConnectionOnCheckin}" />

<property name="acquireRetryAttempts" value="${c3p0.acquireRetryAttempts}" />

<property name="acquireRetryDelay" value="${c3p0.acquireRetryDelay}" />

<property name="breakAfterAcquireFailure" value="${c3p0.breakAfterAcquireFailure}" /> -->

</bean>

<!-- This is the lazy DataSource proxy that interacts with the target DataSource once a real statement is sent to the database. Users use this DataSource to set up their Hibernate session factory, which in turn forces the Hibernate second-level cache and also everything that interacts with that Hibernate session factory to use it. -->

<bean id=*"dataSource"* class=*"org.springframework.jdbc.datasource.LazyConnectionDataSourceProxy"*>

<property name=*"targetDataSource"*><ref bean=*"dataSourceInternal"* /></property>

</bean>

<!-- <bean id="hibSessionFactory" class="org.springframework.orm.hibernate3.annotation.AnnotationSessionFactoryBean"> -->

<bean id=*"hibSessionFactory"* class=*"org.springframework.orm.hibernate4.LocalSessionFactoryBean"*>

<property name=*"dataSource"* ref=*"dataSource"* />

<!--<property name="hibernateProperties"> <value> hibernate.show\_sql=true

</value> </property> -->

<property name=*"hibernateProperties"*>

<value>

hibernate.id.new\_generator\_mappings=true,

hibernate.show\_sql=true

<!-- hibernate.hbm2ddl.auto=update -->

<!-- hibernate.current\_session\_context\_class=thread -->

</value>

</property>

<property name=*"annotatedClasses"*>

<list>

<value>com.ddlab.rnd.orm.Person</value>

<value>com.ddlab.rnd.orm.Employee</value>

<value>com.ddlab.rnd.orm.Owner</value>

</list>

</property>

</bean>

</beans>

# Java Source Code

## Person.java

**package** com.ddlab.rnd.orm;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.Id;

**import** javax.persistence.Inheritance;

**import** javax.persistence.InheritanceType;

**import** javax.persistence.Table;

@Entity

@Table(name = "PERSON3")

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

**public** **class** Person {

@Id

@GeneratedValue

@Column(name = "PERSON\_ID")

**private** Long personId;

@Column(name = "NAME")

**private** String name;

**public** Person() {

}

**public** Person(String name) {

**this**.name = name;

}

**public** Long getPersonId() {

**return** personId;

}

**public** **void** setPersonId(Long personId) {

**this**.personId = personId;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

}

## Employee.java

**package** com.ddlab.rnd.orm;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.PrimaryKeyJoinColumn;

**import** javax.persistence.Table;

@Entity

@Table(name="EMPLOYEE3")

@PrimaryKeyJoinColumn(name="PERSON\_ID")

**public** **class** Employee **extends** Person {

@Column(name="deptName")

**private** String deptName;

**public** Employee() {

}

**public** Employee(String name, String deptName) {

**super**(name);

**this**.deptName = deptName;

}

**public** String getDeptName() {

**return** deptName;

}

**public** **void** setDeptName(String deptName) {

**this**.deptName = deptName;

}

}

## Owner.java

**package** com.ddlab.rnd.orm;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.PrimaryKeyJoinColumn;

**import** javax.persistence.Table;

@Entity

@Table(name="OWNER3")

@PrimaryKeyJoinColumn(name="PERSON\_ID")

**public** **class** Owner **extends** Person {

@Column(name="city")

**private** String city;

**public** Owner() {

}

**public** Owner(String name, String city) {

**super**(name);

**this**.city = city;

}

**public** String getCity() {

**return** city;

}

**public** **void** setCity(String city) {

**this**.city = city;

}

}

## App.java (Test Program)

**package** com.ddlab.rnd.orm;

**import** org.hibernate.HibernateException;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**public** **class** App

{

**public** **static** **void** main( String[] args )

{

ApplicationContext context = **new** ClassPathXmlApplicationContext(

"app-context.xml");

SessionFactory sessionFactory = (SessionFactory) context

.getBean("hibSessionFactory");

Session session = sessionFactory.openSession();

Transaction transaction = **null**;

**try** {

transaction = session.beginTransaction();

Person person = **new** Person("Deb");

session.save(person);

Employee employee = **new** Employee("Deb", "Technology");

session.save(employee);

Owner owner = **new** Owner("Deb", "Bangalore");

session.save(owner);

transaction.commit();

}

**catch** (HibernateException e) {

**if** (transaction != **null**)

transaction.rollback();

e.printStackTrace();

} **finally** {

session.close();

}

System.***out***.println("successfully saved into database");

}

}

## 

Inheritance Mapping in Hibernate (Table Per Concrete Class)

# Maven Configuration (pom.xml)

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>table-per-concrete-class</groupId>

<artifactId>table-per-concrete-class</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>many-to-many</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<spring.version>4.1.2.RELEASE</spring.version>

<spring.security.version>3.2.3.RELEASE</spring.security.version>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.0</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.1.30</version>

</dependency>

<!-- C3P0 library -->

<dependency>

<groupId>com.mchange</groupId>

<artifactId>c3p0</artifactId>

<version>0.9.5</version>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>4.3.7.Final</version>

</dependency>

</dependencies>

</project>

# SQL Configuration (ddl.sql)

**drop** **table** if **exists** owner3;

**drop** **table** if **exists** employee3;

**drop** **table** if **exists** person3;

**CREATE** **TABLE** person3 (

person\_id BIGINT(20) **NOT** **NULL** AUTO\_INCREMENT,

name **VARCHAR**(50) **NOT** **NULL**,

**PRIMARY** **KEY** (person\_id)

);

**CREATE** **TABLE** employee3 (

person\_id BIGINT(10) **NOT** **NULL**,

deptName **VARCHAR**(50) **NULL** **DEFAULT** **NULL**,

**PRIMARY** **KEY** (person\_id),

**FOREIGN** **KEY** (person\_id) **REFERENCES** person3 (person\_id)

);

**CREATE** **TABLE** owner3 (

person\_id BIGINT(20) **NOT** **NULL**,

city **VARCHAR**(50) **NULL** **DEFAULT** **NULL**,

**PRIMARY** **KEY** (person\_id),

**FOREIGN** **KEY** (person\_id) **REFERENCES** person3 (person\_id)

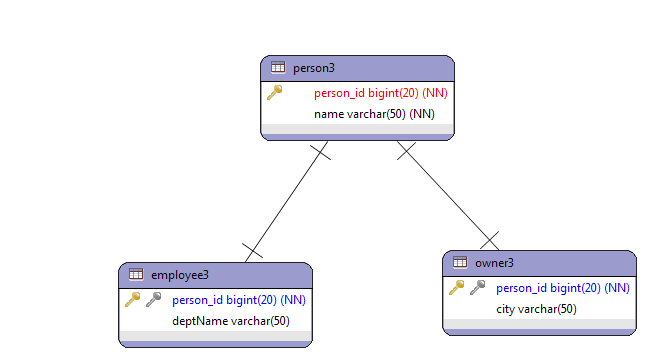
);

**select** \* **from** owner3;

**select** \* **from** employee3;

**select** \* **from** person3;

# Data Model



# Java source code

## Person.java

**package** com.ddlab.rnd.orm;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.Inheritance;

**import** javax.persistence.InheritanceType;

**import** javax.persistence.Table;

@Entity

@Table(name = "PERSON4")

//@Inheritance(strategy = InheritanceType.TABLE\_PER\_CLASS)

**@Inheritance(strategy = InheritanceType.*TABLE\_PER\_CLASS*)**

**public** **class** Person {

@Id

// @GeneratedValue (strategy=GenerationType.IDENTITY)

// @GeneratedValue

// @GeneratedValue (strategy=GenerationType.TABLE)

// @GeneratedValue(strategy = GenerationType.SEQUENCE)

// @GeneratedValue (strategy=GenerationType.AUTO)

@Column(name = "PERSON\_ID")

**private** Long personId;

@Column(name = "name")

**private** String name;

**public** Person() {

}

**public** Person(String name) {

**this**.name = name;

}

**public** Long getPersonId() {

**return** personId;

}

**public** **void** setPersonId(Long personId) {

**this**.personId = personId;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

}

## Employee.java

**package** com.ddlab.rnd.orm;

**import** javax.persistence.AttributeOverride;

**import** javax.persistence.AttributeOverrides;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.Table;

@Entity

@Table(name="EMPLOYEE4")

**@AttributeOverrides({**

**@AttributeOverride(name="name", column=@Column(name="name"))**

**})**

**public** **class** Employee **extends** Person {

@Column(name="deptName")

**private** String deptName;

**public** Employee() {

}

**public** Employee(String name, String deptName) {

**super**(name);

**this**.deptName = deptName;

}

**public** String getDeptName() {

**return** deptName;

}

**public** **void** setDeptName(String deptName) {

**this**.deptName = deptName;

}

}

## Owner.java

**package** com.ddlab.rnd.orm;

**import** javax.persistence.AttributeOverride;

**import** javax.persistence.AttributeOverrides;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.Table;

@Entity

@Table(name="OWNER4")

**@AttributeOverrides({**

**@AttributeOverride(name="name", column=@Column(name="name"))**

**})**

**public** **class** Owner **extends** Person {

@Column(name="city")

**private** String city;

**public** Owner() {

}

**public** Owner(String name, String city) {

**super**(name);

**this**.city = city;

}

**public** String getCity() {

**return** city;

}

**public** **void** setCity(String city) {

**this**.city = city;

}

}

## App.java (Test Program)

**package** com.ddlab.rnd.orm;

**import** org.hibernate.HibernateException;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**public** **class** App

{

**public** **static** **void** main( String[] args )

{

ApplicationContext context = **new** ClassPathXmlApplicationContext(

"app-context.xml");

SessionFactory sessionFactory = (SessionFactory) context

.getBean("hibSessionFactory");

Session session = sessionFactory.openSession();

Transaction transaction = **null**;

**try** {

transaction = session.beginTransaction();

Person person = **new** Person("Deb");

person.setPersonId(5L);

session.save(person);

Employee employee = **new** Employee("Deb", "Technology");

employee.setPersonId(6L);

session.save(employee);

Owner owner = **new** Owner("Deb", "Bangalore");

owner.setPersonId(7L);

session.save(owner);

transaction.commit();

}

**catch** (HibernateException e) {

**if** (transaction != **null**)

transaction.rollback();

e.printStackTrace();

} **finally** {

session.close();

}

System.***out***.println("successfully saved into database");

}

}