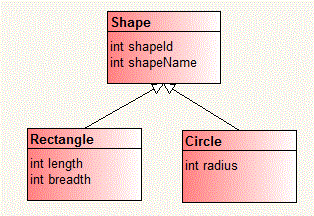
**Inheritance in hibernate:**

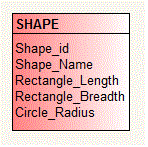
Java is object oriented language and inheritance is one of main functionalities of java.Relation model can implement "is a" and "has a" relationship but hibernate provides us way to implement class hierarchy in a different ways.

**One table per class hierarchy:**

Lets say we have following class hierarchy.We have shape class as base class and Rectangle and Circle inherit from Shape class.

[[](http://1.bp.blogspot.com/-_1V_jaIYZQc/UQ53rNWkypI/AAAAAAAAArE/arvee4TwoVo/s1600/ShapeInheritenceExample.gif)](http://1.bp.blogspot.com/-_1V_jaIYZQc/UQ53rNWkypI/AAAAAAAAArE/arvee4TwoVo/s1600/ShapeInheritenceExample.gif)

In one table per class hierarchy,One table will be created for above hierarchy.i.e. SHAPE table will be created having following structure.

[[](http://4.bp.blogspot.com/-OfUb0uG4UzU/UQ_jBjeEveI/AAAAAAAAAsM/yTnIhX5VFBE/s1600/ShapeInheritenceExample.gif)](http://4.bp.blogspot.com/-OfUb0uG4UzU/UQ_jBjeEveI/AAAAAAAAAsM/yTnIhX5VFBE/s1600/ShapeInheritenceExample.gif)

As you can see only one table(SHAPE) is created having attributes of subclasses also.  
As per our above class diagram,we will create three classes-Shape.java,Rectangle.java and Circle.java

**1.Shape.java**

This is our root class of entity class hierarchy.  
Create **Shape.java**in src->org.arpit.javapostsforlearning.

[view plainprint?](http://www.java2blog.com/2013/02/hibernate-inheritancetable-per-class.html)

1. **package** org.arpit.javapostsforlearning;
2. **import** javax.persistence.Column;
3. **import** javax.persistence.DiscriminatorColumn;
4. **import** javax.persistence.DiscriminatorValue;
5. **import** javax.persistence.Entity;
6. **import** javax.persistence.GeneratedValue;
7. **import** javax.persistence.Id;
8. **import** javax.persistence.Inheritance;
9. **import** javax.persistence.InheritanceType;
10. **import** javax.persistence.Table;
11. **import** javax.persistence.DiscriminatorType;
13. @Entity
14. @Table(name="SHAPE")
15. @Inheritance(strategy=InheritanceType.SINGLE\_TABLE)
16. @DiscriminatorColumn
17. (
18. name="Discriminator",
19. discriminatorType=DiscriminatorType.STRING
20. )
21. @DiscriminatorValue(value="S")
22. **public** **class** Shape {
24. @Id
25. @GeneratedValue
26. @Column(name="Shape\_Id")
27. **int** shapeId;
28. @Column(name="Shape\_Name")
29. String shapeName;
31. **public** Shape()
32. {
34. }
35. **public** Shape(String shapeName)
36. {
37. **this**.shapeName=shapeName;
38. }
39. //getters and setters
41. }

Shape is our root class so some annotations needs to be used with root class for implementing inheritance.

**@Inheritance:**

For implementing inheritance in hiberante,@Inheritance annotation is used.It defines inheritance strategy to be implement for entity class hierarchy.For one table per class hierarhcy,we have used Single\_Table as inheritance strategy.This annotation is defined at root level or sub hierarchy level where different strategy is to be applied.

**@DiscriminatorColumn:**

This annotation is used to define discriminator column for Single\_Table and joined strategy.It is used to distinguish between different class instances.This annotation is defined at root level or sub hierarchy level where different strategy is to be applied.

If @DiscriminatorColumn annotation is not specified,then hibernate will create a column named as "DType" and DiscriminatorType will be string.

**@DiscriminatorValue:**

This annotation defines value in discriminator column for that class.This can only be applied on entity concrete class.For example,If entry will be of shape instance in SHAPE table then "s" will be value for that row in discriminator column.If this annotation is not specified and Discriminator column is used then provider specific values will be provided and if Discriminator type is String then discriminator value will be entity name.Discriminator value,if not defaulted need to specified on each enitity in hierarchy.

**2.Rectangle.java**

This is our child class.  
Create **Rectangle.java**in src->org.arpit.javapostsforlearning.

[view plainprint?](http://www.java2blog.com/2013/02/hibernate-inheritancetable-per-class.html)

1. **package** org.arpit.javapostsforlearning;**import** javax.persistence.Column;
2. **import** javax.persistence.Column;
3. **import** javax.persistence.DiscriminatorValue;
4. **import** javax.persistence.Entity;
6. @Entity
7. @DiscriminatorValue(value="R")
8. **public** **class** Rectangle **extends** Shape{
10. @Column(name="Rectangle\_Length")
11. **int** length;
12. @Column(name="Rectangle\_Breadth")
13. **int** breadth;
14. // getters and setters
16. **public** Rectangle()
17. {
19. }
21. **public** Rectangle(String shapeName,**int** length,**int** breadth)
22. {
23. **super**(shapeName);
24. **this**.length=length;
25. **this**.breadth=breadth;
26. }
28. // getters and setters
29. }

**3.Circle.java**

This is our second child class.

Create **Circle** **.java**in src->org.arpit.javapostsforlearning.

[view plainprint?](http://www.java2blog.com/2013/02/hibernate-inheritancetable-per-class.html)

1. **package** org.arpit.javapostsforlearning;**import** javax.persistence.Column;
2. **import** javax.persistence.Column;
3. **import** javax.persistence.DiscriminatorValue;
4. **import** javax.persistence.Entity;
6. @Entity
7. @DiscriminatorValue(value="C")
8. **public** **class** Circle **extends** Shape{
10. @Column(name="Circle\_Radius")
11. **int** radius;
13. **public** Circle()
14. {
16. }
17. **public** Circle(String shapeName,**int** radius)
18. {
19. **super**(shapeName);
20. **this**.radius=radius;
22. }
23. // getters and setters
24. }

**4.Hiberante.cfg.xml:**

Create a file named "hibernate.cfg.xml" in **src** folder.

[view plainprint?](http://www.java2blog.com/2013/02/hibernate-inheritancetable-per-class.html)

1. **<?xml** version='1.0' encoding='utf-8'**?>**
2. <!DOCTYPE hibernate-configuration PUBLIC
3. "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
4. "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd"**>**
6. **<hibernate-configuration>**
8. **<session-factory>**
10. <!-- Database connection settings -->
11. **<property** name="connection.driver\_class"**>**com.microsoft.sqlserver.jdbc.SQLServerDriver**</property>**
12. **<property** name="connection.url"**>**jdbc:sqlserver://localhost:1433;database=UserInfo**</property>**
13. **<property** name="connection.username"**>**sa**</property>**
14. **<property** name="connection.password"**></property>**
16. <!-- JDBC connection pool (use the built-in) -->
17. **<property** name="connection.pool\_size"**>**1**</property>**
19. <!-- SQL dialect -->
20. **<property** name="dialect"**>**org.hibernate.dialect.SQLServer2005Dialect**</property>**
22. <!-- Enable Hibernate's automatic session context management -->
23. **<property** name="current\_session\_context\_class"**>**thread**</property>**
25. <!-- Disable the second-level cache  -->
26. **<property** name="cache.provider\_class"**>**org.hibernate.cache.NoCacheProvider**</property>**
28. <!-- Echo all executed SQL to stdout -->
29. **<property** name="show\_sql"**>**true**</property>**
31. <!-- Drop and re-create the database schema on startup -->
32. **<property** name="hbm2ddl.auto"**>**create**</property>**
34. **<mapping** class="org.arpit.javapostsforlearning.Shape"**></mapping>**
35. **<mapping** class="org.arpit.javapostsforlearning.Rectangle"**></mapping>**
36. **<mapping** class="org.arpit.javapostsforlearning.Circle"**></mapping>**
38. **</session-factory>**
40. **</hibernate-configuration>**

**5.Main Class:**

[view plainprint?](http://www.java2blog.com/2013/02/hibernate-inheritancetable-per-class.html)

1. **package** org.arpit.javapostsforlearning;
3. **import** org.hibernate.Session;
4. **import** org.hibernate.SessionFactory;
5. **import** org.hibernate.cfg.Configuration;
6. **import** org.hibernate.service.ServiceRegistry;
7. **import** org.hibernate.service.ServiceRegistryBuilder;
9. **public** **class** HibernateMain {
11. **public** **static** **void** main(String[] args) {
13. Shape shape=**new** Shape("Sqaure");
14. Rectangle rectangle=**new** Rectangle("Rectangle", 10, 20);
15. Circle circle=**new** Circle("Circle", 4);
17. Configuration configuration=**new** Configuration();
18. configuration.configure();
19. ServiceRegistry sr= **new** ServiceRegistryBuilder().applySettings(configuration.getProperties()).buildServiceRegistry();
20. SessionFactory sf=configuration.buildSessionFactory(sr);
21. Session ss=sf.openSession();
23. ss.beginTransaction();
24. ss.save(shape);
25. ss.save(rectangle);
26. ss.save(circle);
27. ss.getTransaction().commit();
28. ss.close();
30. }
31. }

**Project structure:**

**6.Run it:**

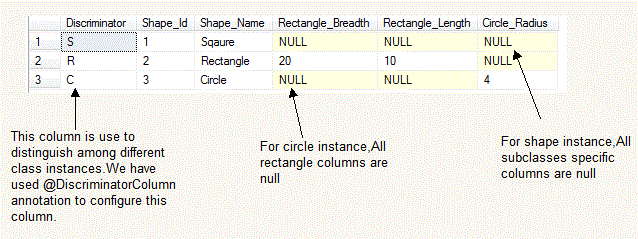
When you run it,you will get following output.

[view plainprint?](http://www.java2blog.com/2013/02/hibernate-inheritancetable-per-class.html)

1. Hibernate: create table SHAPE (Discriminator varchar(31) not **null**, Shape\_Id **int** identity not **null**, Shape\_Name varchar(255), Rectangle\_Breadth **int**, Rectangle\_Length **int**, Circle\_Radius **int**, primary key (Shape\_Id))
2. Feb 04, 2013 11:01:36 PM org.hibernate.tool.hbm2ddl.SchemaExport execute
3. INFO: HHH000230: Schema export complete
4. Hibernate: insert into SHAPE (Shape\_Name, Discriminator) values (?, 'S')
5. Hibernate: insert into SHAPE (Shape\_Name, Rectangle\_Breadth, Rectangle\_Length, Discriminator) values (?, ?, ?, 'R')
6. Hibernate: insert into SHAPE (Shape\_Name, Circle\_Radius, Discriminator) values (?, ?, 'C')

**7.SQL output:**

SHAPE table in database.

[[](http://1.bp.blogspot.com/-vbZ-GtCjUa0/UQ_8CuhL0GI/AAAAAAAAAs0/iUNLV680QZc/s1600/InheritanceClassSQLOutput.gif)](http://1.bp.blogspot.com/-vbZ-GtCjUa0/UQ_8CuhL0GI/AAAAAAAAAs0/iUNLV680QZc/s1600/InheritanceClassSQLOutput.gif)

Read more at http://www.java2blog.com/2013/02/hibernate-inheritancetable-per-class.html#XokVxhlEe1smV60H.99