## **JPA Mapping Annotations**

1. @Access  
2. @AssociationOverride  
3. @AssociationOverrides  
4. @AttributeOverride  
5. @AttributeOverrides  
6. @Basic  
7. @Cacheable  
8. @CollectionTable  
9. @Column  
10. @ColumnResult  
11. @ConstructorResult  
12. @Convert  
13. @Converter  
14. @Converts  
15. @DiscriminatorColumn  
16. @DiscriminatorValue  
17. @ElementCollection  
18. @Embeddable  
19. @Embedded  
20. @EmbeddedId  
21. @Entity  
22. @EntityListeners  
23. @EntityResult  
24. @Enumerated  
25. @ExcludeDefaultListeners  
26. @ExcludeSuperclassListeners  
27. @FieldResult  
28. @ForeignKey  
29. @GeneratedValue  
30. @Id  
31. @IdClass  
32. @Index  
33. @Inheritance  
34. @JoinColumn  
35. @JoinColumns  
36. @JoinTable  
37. @Lob  
38. @ManyToMany  
39. @ManyToOne  
40. @MapKey  
41. @MapKeyClass  
42. @MapKeyColumn  
43. @MapKeyEnumerated  
44. @MapKeyJoinColumn  
45. @MapKeyJoinColumns  
46. @MapKeyTemporal  
47. @MappedSuperclass  
48. @MapsId  
49. @NamedAttributeNode  
50. @NamedEntityGraph  
51. @NamedEntityGraphs  
52. @NamedNativeQueries  
53. @NamedNativeQuery  
54. @NamedQueries  
55. @NamedQuery  
56. @NamedStoredProcedureQueries  
57. @NamedStoredProcedureQuery  
58. @NamedSubgraph  
59. @OneToMany  
60. @OneToOne  
61. @OrderBy  
62. @OrderColumn  
63. @PersistenceContext  
64. @PersistenceContexts  
65. @PersistenceProperty  
66. @PersistenceUnit  
67. @PersistenceUnits  
68. @PostLoad  
69. @PostPersist  
70. @PostRemove  
71. @PostUpdate  
72. @PrePersist  
73. @PreRemove  
74. @PreUpdate  
75. @PrimaryKeyJoinColumn  
76. @PrimaryKeyJoinColumns  
77. @QueryHint  
78. @SecondaryTable  
79. @SecondaryTables  
80. @SequenceGenerator  
81. @SqlResultSetMapping  
82. @SqlResultSetMappings  
83. @StoredProcedureParameter  
84. @Table  
85. @TableGenerator  
86. @Temporal  
87. @Transient  
88. @UniqueConstraint  
89. @Version

## **Hibernate Mapping Annotations**

1. @AccessType
2. @Any
3. @AnyMetaDef
4. @AnyMetaDefs
5. @AttributeAccessor
6. @BatchSize
7. @Cache
8. @Cascade
9. @Check
10. @CollectionId
11. @CollectionType
12. @ColumnDefault
13. @Columns
14. @ColumnTransformer
15. @ColumnTransformers
16. @CreationTimestamp
17. @DiscriminatorFormula
18. @DiscriminatorOptions
19. @DynamicInsert
20. @DynamicUpdate
21. @Entity @Fetch
22. @FetchProfile
23. @FetchProfile.FetchOverride
24. @FetchProfiles
25. @Filter
26. @FilterDef
27. @FilterDefs
28. @FilterJoinTable
29. @FilterJoinTables
30. @Filters
31. @ForeignKey
32. @Formula
33. @Generated
34. @GeneratorType
35. @GenericGenerator
36. @GenericGenerators
37. @Immutable
38. @Index
39. @IndexColumn
40. @JoinColumnOrFormula
41. @JoinColumnsOrFormulas
42. @JoinFormula
43. @LazyCollection
44. @LazyGroup
45. @LazyToOne
46. @ListIndexBase
47. @Loader
48. @ManyToAny
49. @MapKeyType
50. @MetaValue
51. @NamedNativeQueries
52. @NamedQueries
53. @NamedQuery
54. @Nationalized
55. @NaturalId
56. @NaturalIdCache
57. @NotFound
58. @OnDelete
59. @OptimisticLock
60. @OptimisticLocking
61. @OrderBy
62. @ParamDef
63. @Parameter
64. @Parent
65. @Persister
66. @Polymorphism
67. @Proxy
68. @RowId
69. @SelectBeforeUpdate
70. @Sort
71. @SortComparator
72. @SortNatural
73. @Source
74. @SQLDelete
75. @SQLDeleteAll
76. @SqlFragmentAlias
77. @SQLInsert
78. @SQLUpdate
79. @Subselect
80. @Synchronize
81. @Table
82. @Tables
83. @Target
84. @Tuplizer
85. @Tuplizers
86. @Type
87. @TypeDef
88. @TypeDefs
89. @UpdateTimestamp
90. @ValueGenerationType
91. @Where
92. @WhereJoinTable

## **JPA Mapping Annotations**

## **@Access**

The [@Access](http://docs.oracle.com/javaee/7/api/javax/persistence/Access.html) annotation is used to specify the access type of the associated entity class, mapped superclass, or embeddable class, or entity attribute.  
See the [Access type](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#access) section for more info.

## **@AssociationOverride**

The [@AssociationOverride](http://docs.oracle.com/javaee/7/api/javax/persistence/AssociationOverride.html) annotation is used to override an association mapping (e.g. @ManyToOne, @OneToOne, @OneToMany, @ManyToMany) inherited from a mapped superclass or an embeddable.  
See the [Overriding Embeddable types](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#embeddable-override) section for more info.

## **@AssociationOverrides**

The [@AssociationOverrides](http://docs.oracle.com/javaee/7/api/javax/persistence/AssociationOverrides.html) is used to group several [@AssociationOverride](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-associationoverride) annotations.

## **@AttributeOverride**

The [@AttributeOverride](http://docs.oracle.com/javaee/7/api/javax/persistence/AttributeOverride.html) annotation is used to override an attribute mapping inherited from a mapped superclass or an embeddable.  
See the [Overriding Embeddable types](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#embeddable-override) section for more info.

## **@AttributeOverrides**

The [@AttributeOverrides](http://docs.oracle.com/javaee/7/api/javax/persistence/AttributeOverrides.html) is used to group several [@AttributeOverride](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-attributeoverride) annotations.

## **@Basic**

The [@Basic](http://docs.oracle.com/javaee/7/api/javax/persistence/Basic.html) annotation is used to map a basic attribute type to a database column.  
See the [Basic types](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#basic) chapter for more info.

## **@Cacheable**

The [@Cacheable](http://docs.oracle.com/javaee/7/api/javax/persistence/Cacheable.html) annotation is used to specify whether an entity should be stored in the second-level cache.  
See the [Caching](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#caching) chapter for more info.

## **@CollectionTable**

The [@CollectionTable](http://docs.oracle.com/javaee/7/api/javax/persistence/CollectionTable.html) annotation is used to specify the database table that stores the values of a basic or an embeddable type collection.  
See the [Collections of embeddable types](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#embeddable-collections) section for more info.

## **@Column**

The [@Column](http://docs.oracle.com/javaee/7/api/javax/persistence/Column.html) annotation is used to specify the mapping between a basic entity attribute and the database table column.  
See the [@Column annotation](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#basic-column-annotation) section for more info.

## **@ColumnResult**

The [@ColumnResult](http://docs.oracle.com/javaee/7/api/javax/persistence/ColumnResult.html) annotation is used in conjunction with the [@SqlResultSetMapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-sqlresultsetmapping) or [@ConstructorResult](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-constructorresult)annotations to map a SQL column for a given SELECT query.  
See the [Entity associations with named native queries](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#sql-composite-key-entity-associations_named-query-example) section for more info.

## **@ConstructorResult**

The [@ConstructorResult](http://docs.oracle.com/javaee/7/api/javax/persistence/ConstructorResult.html) annotation is used in conjunction with the [@SqlResultSetMapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-sqlresultsetmapping) annotations to map columns of a given SELECT query to a certain object constructor.  
See the [Multiple scalar values NamedNativeQuery with ConstructorResult](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#sql-multiple-scalar-values-dto-NamedNativeQuery-example) section for more info.

## **@Convert**

The [@Convert](http://docs.oracle.com/javaee/7/api/javax/persistence/Convert.html) annotation is used to specify the [AttributeConverter](http://docs.oracle.com/javaee/7/api/javax/persistence/AttributeConverter.html) implementation used to convert the currently annotated basic attribute.  
See the [AttributeConverter](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#basic-enums-attribute-converter) section for more info.

## **24.1.13. @Converter**

The [@Converter](http://docs.oracle.com/javaee/7/api/javax/persistence/Converter.html) annotation is used to specify that the current annotate [AttributeConverter](http://docs.oracle.com/javaee/7/api/javax/persistence/AttributeConverter.html) implementation can be used as a JPA basic attribute converter.  
See the [AttributeConverter](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#basic-enums-attribute-converter) section for more info.

## **@Converts**

The [@Converts](http://docs.oracle.com/javaee/7/api/javax/persistence/Converts.html) annotation is used to group multiple [@Convert](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-convert) annotations.  
See the [AttributeConverter](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#basic-enums-attribute-converter) section for more info.

## **@DiscriminatorColumn**

The [@DiscriminatorColumn](http://docs.oracle.com/javaee/7/api/javax/persistence/DiscriminatorColumn.html) annotation is used to specify the discriminator column name and the [discriminator type](http://docs.oracle.com/javaee/7/api/javax/persistence/DiscriminatorColumn.html#discriminatorType--) for the SINGLE\_TABLE and JOINED Inheritance strategies.  
See the [Discriminator](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#entity-inheritance-discriminator) section for more info.

## **@DiscriminatorValue**

The [@DiscriminatorValue](http://docs.oracle.com/javaee/7/api/javax/persistence/DiscriminatorValue.html) annotation is used to specify what value of the discriminator column is used for mapping the currently annotated entity.  
See the [Discriminator](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#entity-inheritance-discriminator) section for more info.

## **@ElementCollection**

The [@ElementCollection](http://docs.oracle.com/javaee/7/api/javax/persistence/ElementCollection.html) annotation is used to specify a collection of a basic or embeddable types.  
See the [Collections](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#collections) section for more info.

## **@Embeddable**

The [@Embeddable](http://docs.oracle.com/javaee/7/api/javax/persistence/Embeddable.html) annotation is used to specify embeddable types. Like basic types, embeddable types do not have any identity, being managed by their owning entity.  
See the [Embeddables](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#embeddables) section for more info.

## **@Embedded**

The [@Embedded](http://docs.oracle.com/javaee/7/api/javax/persistence/Embedded.html) annotation is used to specify that a given entity attribute represents an embeddable type.  
See the [Embeddables](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#embeddables) section for more info.

## **@EmbeddedId**

The [@EmbeddedId](http://docs.oracle.com/javaee/7/api/javax/persistence/EmbeddedId.html) annotation is used to specify the entity identifier is an embeddable type.  
See the [Composite identifiers with @EmbeddedId](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#identifiers-composite-aggregated) section for more info.

## **@Entity**

The [@Entity](http://docs.oracle.com/javaee/7/api/javax/persistence/Entity.html) annotation is used to specify that the currently annotate class represents an entity type. Unlike basic and embeddable types, entity types have an identity and their state is managed by the underlying Persistence Context.  
See the [Entity](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#entity) section for more info.

## **@EntityListeners**

The [@EntityListeners](http://docs.oracle.com/javaee/7/api/javax/persistence/EntityListeners.html) annotation is used to specify an array of callback listener classes that are used by the currently annotated entity.  
See the [JPA callbacks](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#events-jpa-callbacks-example) section for more info.

## **@EntityResult**

The [@EntityResult](http://docs.oracle.com/javaee/7/api/javax/persistence/EntityResult.html) annotation is used with the [@SqlResultSetMapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-sqlresultsetmapping) annotation to map the selected columns to an entity.  
See the [Entity associations with named native queries](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#sql-composite-key-entity-associations_named-query-example) section for more info.

## **@Enumerated**

The [@Enumerated](http://docs.oracle.com/javaee/7/api/javax/persistence/Enumerated.html) annotation is used to specify that an entity attribute represents an enumerated type.  
See the [@Enumerated basic type](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#basic-enums-Enumerated) section for more info.

## **@ExcludeDefaultListeners**

The [@ExcludeDefaultListeners](http://docs.oracle.com/javaee/7/api/javax/persistence/ExcludeDefaultListeners.html) annotation is used to specify that the currently annotated entity skips the invocation of any default listener.  
See the [Exclude default entity listeners](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#events-exclude-default-listener) section for more info.

## **@ExcludeSuperclassListeners**

The [@ExcludeSuperclassListeners](http://docs.oracle.com/javaee/7/api/javax/persistence/ExcludeSuperclassListeners.html) annotation is used to specify that the currently annotated entity skips the invocation of listeners declared by its superclass.  
See the [Exclude default entity listeners](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#events-exclude-default-listener) section for more info.

## **@FieldResult**

The [@FieldResult](http://docs.oracle.com/javaee/7/api/javax/persistence/FieldResult.html) annotation is used with the [@EntityResult](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-entityresult) annotation to map the selected columns to the fields of some specific entity.  
See the [Entity associations with named native queries](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#sql-composite-key-entity-associations_named-query-example) section for more info.

## **@ForeignKey**

The [@ForeignKey](http://docs.oracle.com/javaee/7/api/javax/persistence/ForeignKey.html) annotation is used to specify the associated foreign key of a [@JoinColumn](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-joincolumn) mapping. The @ForeignKeyannotation is only used if the automated schema generation tool is enabled, in which case, it allows you to customize the underlying foreign key definition.  
See the [@ManyToOne with @ForeignKey](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#associations-many-to-one-example) section for more info.

## **@GeneratedValue**

The [@GeneratedValue](http://docs.oracle.com/javaee/7/api/javax/persistence/GeneratedValue.html) annotation specifies that the entity identifier value is automatically generated using an identity column, a database sequence, or a table generator. Hibernate supports the @GeneratedValue mapping even for UUIDidentifiers.  
See the [Automatically-generated identifiers](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#identifiers-simple-generated) section for more info.

## **@Id**

The [@Id](http://docs.oracle.com/javaee/7/api/javax/persistence/Id.html) annotation specifies the entity identifier. An entity must always have an identifier attribute which is used when loading the entity in a given Persistence Context.  
See the [Identifiers](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#identifiers) section for more info.

## **@IdClass**

The [@IdClass](http://docs.oracle.com/javaee/7/api/javax/persistence/IdClass.html) annotation is used if the current entity defined a composite identifier. A separate class encapsulates all the identifier attributes, which are mirrored by the current entity mapping.  
See the [Composite identifiers with @IdClass](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#identifiers-composite-nonaggregated) section for more info.

## **@Index**

The [@Index](http://docs.oracle.com/javaee/7/api/javax/persistence/Index.html) annotation is used by the automated schema generation tool to create a database index.  
See the [Columns index](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#schema-generation-columns-index) chapter for more info.  
24.1.33. @Inheritance  
The [@Inheritance](http://docs.oracle.com/javaee/7/api/javax/persistence/Inheritance.html) annotation is used to specify the inheritance strategy of a given entity class hierarchy.  
See the [Inheritance](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#entity-inheritance) section for more info.

## **@JoinColumn**

The [@JoinColumn](http://docs.oracle.com/javaee/7/api/javax/persistence/JoinColumn.html) annotation is used to specify the FOREIGN KEY column used when joining an entity association or an embeddable collection.  
See the [@ManyToOne with @JoinColumn](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#associations-many-to-one-example) section for more info.

## **@JoinColumns**

The [@JoinColumns](http://docs.oracle.com/javaee/7/api/javax/persistence/JoinColumns.html) annotation is used to group multiple [@JoinColumn](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-joincolumn) annotations, which are used when mapping entity association or an embeddable collection using a composite identifier

## **@JoinTable**

The [@JoinTable](http://docs.oracle.com/javaee/7/api/javax/persistence/JoinTable.html) annotation is used to specify the link table between two other database tables.  
See the [@JoinTable mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#collections-map-unidirectional-example) section for more info.

## **@Lob**

The [@Lob](http://docs.oracle.com/javaee/7/api/javax/persistence/Lob.html) annotation is used to specify that the currently annotated entity attribute represents a large object type.  
See the [BLOB mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#basic-blob-example) section for more info.

## **@ManyToMany**

The [@ManyToMany](http://docs.oracle.com/javaee/7/api/javax/persistence/ManyToMany.html) annotation is used to specify a many-to-many database relationship.  
See the [@ManyToMany mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#associations-many-to-many) section for more info.

## **@ManyToOne**

The [@ManyToOne](http://docs.oracle.com/javaee/7/api/javax/persistence/ManyToOne.html) annotation is used to specify a many-to-one database relationship.  
See the [@ManyToOne mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#associations-many-to-one) section for more info.

## **@MapKey**

The [@MapKey](http://docs.oracle.com/javaee/7/api/javax/persistence/MapKey.html) annotation is used to specify the key of a java.util.Map association for which the key type is either the primary key or an attribute of the entity which represents the value of the map.  
See the [@MapKey mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#collections-map-unidirectional-example) section for more info.

## **@MapKeyClass**

The [@MapKeyClass](http://docs.oracle.com/javaee/7/api/javax/persistence/MapKeyClass.html) annotation is used to specify the type of the map key of a java.util.Map associations.  
See the [@MapKeyClass mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#collections-map-key-class) section for more info.

## **@MapKeyColumn**

The [@MapKeyColumn](http://docs.oracle.com/javaee/7/api/javax/persistence/MapKeyColumn.html) annotation is used to specify the database column which stores the key of a java.util.Mapassociation for which the map key is a basic type.  
See the [@MapKeyType mapping section](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#collections-map-custom-key-type-mapping-example) for an example of @MapKeyColumn annotation usage.

## **@MapKeyEnumerated**

The [@MapKeyEnumerated](http://docs.oracle.com/javaee/7/api/javax/persistence/MapKeyEnumerated.html) annotation is used to specify that the key of java.util.Map association is a Java Enum.  
See the [@MapKeyEnumerated mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#collections-map-bidirectional-example) section for more info.

## **@MapKeyJoinColumn**

The [@MapKeyJoinColumn](http://docs.oracle.com/javaee/7/api/javax/persistence/MapKeyJoinColumn.html) annotation is used to specify that the key of java.util.Map association is an entity association. The map key column is a FOREIGN KEY in a link table that also joins the Map owner’s table with the table where the Map value resides.  
See the [@MapKeyJoinColumn mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#collections-map-value-type-entity-key-example) section for more info.

## **@MapKeyJoinColumns**

The [@MapKeyJoinColumns](http://docs.oracle.com/javaee/7/api/javax/persistence/MapKeyJoinColumns.html) annotation is used to group several [@MapKeyJoinColumn](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-mapkeyjoincolumn) mappings when the java.util.Map association key uses a composite identifier.

## **@MapKeyTemporal**

The [@MapKeyTemporal](http://docs.oracle.com/javaee/7/api/javax/persistence/MapKeyTemporal.html) annotation is used to specify that the key of java.util.Map association is a [@TemporalType](http://docs.oracle.com/javaee/7/api/javax/persistence/TemporalType.html) (e.g. DATE, TIME, TIMESTAMP).  
See the [@MapKeyTemporal mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#collections-map-unidirectional-example) section for more info.

## **@MappedSuperclass**

The [@MappedSuperclass](http://docs.oracle.com/javaee/7/api/javax/persistence/MappedSuperclass.html) annotation is used to specify that the currently annotated type attributes are inherited by any subclass entity.  
See the [@MappedSuperclass](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#entity-inheritance-mapped-superclass) section for more info.

## **@MapsId**

The [@MapsId](http://docs.oracle.com/javaee/7/api/javax/persistence/MapsId.html) annotation is used to specify that the entity identifier is mapped by the currently annotated @ManyToOne or @OneToOne association.  
See the [@MapsId mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#identifiers-derived-mapsid) section for more info.

## **@NamedAttributeNode**

The [@NamedAttributeNode](http://docs.oracle.com/javaee/7/api/javax/persistence/NamedAttributeNode.html) annotation is used to specify each individual attribute node that needs to be fetched by an Entity Graph.  
See the [Fetch graph](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#fetching-strategies-dynamic-fetching-entity-graph-example) section for more info.

## **@NamedEntityGraph**

The [@NamedEntityGraph](http://docs.oracle.com/javaee/7/api/javax/persistence/NamedEntityGraph.html) annotation is used to specify an Entity Graph that can be used by an entity query to override the default fetch plan.  
See the [Fetch graph](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#fetching-strategies-dynamic-fetching-entity-graph-example) section for more info.

## **@NamedEntityGraphs**

The [@NamedEntityGraphs](http://docs.oracle.com/javaee/7/api/javax/persistence/NamedEntityGraphs.html) annotation is used to group multiple [@NamedEntityGraph](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-namedentitygraph) annotations.  
24.1.52. @NamedNativeQueries  
The [@NamedNativeQueries](http://docs.oracle.com/javaee/7/api/javax/persistence/NamedNativeQueries.html) annotation is used to group multiple [@NamedNativeQuery](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-namednativequery) annotations.  
See the [Custom CRUD mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#sql-custom-crud-example) section for more info.

## **@NamedNativeQuery**

The [@NamedNativeQuery](http://docs.oracle.com/javaee/7/api/javax/persistence/NamedNativeQuery.html) annotation is used to specify a native SQL query that can be retrieved later by its name.  
See the [Custom CRUD mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#sql-custom-crud-example) section for more info.

## **@NamedQueries**

The [@NamedQueries](http://docs.oracle.com/javaee/7/api/javax/persistence/NamedQueries.html) annotation is used to group multiple [@NamedQuery](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-namedquery) annotations.

## **@NamedQuery**

The [@NamedQuery](http://docs.oracle.com/javaee/7/api/javax/persistence/NamedQuery.html) annotation is used to specify a JPQL query that can be retrieved later by its name.  
See the [@NamedQuery](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#jpql-api-named-query-example) section for more info.

## **@NamedStoredProcedureQueries**

The [@NamedStoredProcedureQueries](http://docs.oracle.com/javaee/7/api/javax/persistence/NamedStoredProcedureQueries.html) annotation is used to group multiple [@NamedStoredProcedureQuery](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-namedstoredprocedurequery)annotations.

## **@NamedStoredProcedureQuery**

The [@NamedStoredProcedureQuery](http://docs.oracle.com/javaee/7/api/javax/persistence/NamedStoredProcedureQuery.html) annotation is used to specify a stored procedure query that can be retrieved later by its name.  
See the [Using named queries to call stored procedures](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#sql-sp-named-query) section for more info.

## **@NamedSubgraph**

The [@NamedSubgraph](http://docs.oracle.com/javaee/7/api/javax/persistence/NamedSubgraph.html) annotation used to specify a subgraph in an Entity Graph.  
See the [Fetch subgraph](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#fetching-strategies-dynamic-fetching-entity-subgraph) section for more info.

## **@OneToMany**

The [@OneToMany](http://docs.oracle.com/javaee/7/api/javax/persistence/OneToMany.html) annotation is used to specify a one-to-many database relationship.  
See the [@OneToMany mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#associations-one-to-many) section for more info.

## **@OneToOne**

The [@OneToOne](http://docs.oracle.com/javaee/7/api/javax/persistence/OneToOne.html) annotation is used to specify a one-to-one database relationship.  
See the [@OneToOne mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#associations-one-to-one) section for more info.

## **@OrderBy**

The [@OrderBy](http://docs.oracle.com/javaee/7/api/javax/persistence/OrderBy.html) annotation is used to specify the entity attributes used for sorting when fetching the currently annotated collection.  
See the [@OrderBy mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#collections-unidirectional-ordered-list) section for more info.

## **@OrderColumn**

The [@OrderColumn](http://docs.oracle.com/javaee/7/api/javax/persistence/OrderColumn.html) annotation is used to specify that the current annotation collection order should be materialized in the database.  
See the [@OrderColumn mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#collections-unidirectional-ordered-list-order-column-example) section for more info.

## **@PersistenceContext**

The [@PersistenceContext](http://docs.oracle.com/javaee/7/api/javax/persistence/PersistenceContext.html) annotation is used to specify the EntityManager that needs to be injected as a dependency.  
See the [@PersistenceContext mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#bootstrap-jpa-compliant-PersistenceContext-example) section for more info.

## **@PersistenceContexts**

The [@PersistenceContexts](http://docs.oracle.com/javaee/7/api/javax/persistence/PersistenceContexts.html) annotation is used to group multiple [@PersistenceContext](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-persistencecontext) annotations.

## **@PersistenceProperty**

The [@PersistenceProperty](http://docs.oracle.com/javaee/7/api/javax/persistence/PersistenceProperty.html) annotation is used by the [@PersistenceContext](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-persistencecontext) annotation to declare JPA provider properties that are passed to the underlying container when the EntityManager instance is created.  
See the [@PersistenceProperty mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#bootstrap-jpa-compliant-PersistenceContext-configurable-example) section for more info.

## **@PersistenceUnit**

The [@PersistenceUnit](http://docs.oracle.com/javaee/7/api/javax/persistence/PersistenceUnit.html) annotation is used to specify the EntityManagerFactory that needs to be injected as a dependency.  
See the [@PersistenceUnit mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#bootstrap-jpa-compliant-PersistenceUnit-example) section for more info.

## **@PersistenceUnits**

The [@PersistenceUnits](http://docs.oracle.com/javaee/7/api/javax/persistence/PersistenceUnits.html) annotation is used to group multiple [@PersistenceUnit](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-persistenceunit) annotations.

## **@PostLoad**

The [@PostLoad](http://docs.oracle.com/javaee/7/api/javax/persistence/PostLoad.html) annotation is used to specify a callback method that fires after an entity is loaded.  
See the [JPA callbacks](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#events-jpa-callbacks-example) section for more info.

## **@PostPersist**

The [@PostPersist](http://docs.oracle.com/javaee/7/api/javax/persistence/PostPersist.html) annotation is used to specify a callback method that fires after an entity is persisted.  
See the [JPA callbacks](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#events-jpa-callbacks-example) section for more info.

## **@PostRemove**

The [@PostRemove](http://docs.oracle.com/javaee/7/api/javax/persistence/PostRemove.html) annotation is used to specify a callback method that fires after an entity is removed.  
See the [JPA callbacks](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#events-jpa-callbacks-example) section for more info.

## **@PostUpdate**

The [@PostUpdate](http://docs.oracle.com/javaee/7/api/javax/persistence/PostUpdate.html) annotation is used to specify a callback method that fires after an entity is updated.  
See the [JPA callbacks](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#events-jpa-callbacks-example) section for more info.

## **@PrePersist**

The [@PrePersist](http://docs.oracle.com/javaee/7/api/javax/persistence/PrePersist.html) annotation is used to specify a callback method that fires before an entity is persisted.  
See the [JPA callbacks](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#events-jpa-callbacks-example) section for more info.

## **@PreRemove**

The [@PreRemove](http://docs.oracle.com/javaee/7/api/javax/persistence/PreRemove.html) annotation is used to specify a callback method that fires before an entity is removed.  
See the [JPA callbacks](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#events-jpa-callbacks-example) section for more info.

## **@PreUpdate**

The [@PreUpdate](http://docs.oracle.com/javaee/7/api/javax/persistence/PreUpdate.html) annotation is used to specify a callback method that fires before an entity is updated.  
See the [JPA callbacks](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#events-jpa-callbacks-example) section for more info.

## **@PrimaryKeyJoinColumn**

The [@PrimaryKeyJoinColumn](http://docs.oracle.com/javaee/7/api/javax/persistence/PrimaryKeyJoinColumn.html) annotation is used to specify that the primary key column of the currently annotated entity is also a foreign key to some other entity (e.g. a base class table in a JOINED inheritance strategy, the primary table in a secondary table mapping, or the parent table in a @OneToOne relationship).  
See the [@PrimaryKeyJoinColumn mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#identifiers-derived-primarykeyjoincolumn) section for more info.

## **@PrimaryKeyJoinColumns**

The [@PrimaryKeyJoinColumns](http://docs.oracle.com/javaee/7/api/javax/persistence/PrimaryKeyJoinColumns.html) annotation is used to group multiple [@PrimaryKeyJoinColumn](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-primarykeyjoincolumn) annotations.

## **@QueryHint**

The [@QueryHint](http://docs.oracle.com/javaee/7/api/javax/persistence/QueryHint.html) annotation is used to specify a JPA provider hint used by a @NamedQuery or a @NamedNativeQueryannotation.  
See the [@QueryHint](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#jpa-read-only-entities-native-example) section for more info.

## **@SecondaryTable**

The [@SecondaryTable](http://docs.oracle.com/javaee/7/api/javax/persistence/SecondaryTable.html) annotation is used to specify a secondary table for the currently annotated entity.  
See the [@SecondaryTable mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#sql-custom-crud-secondary-table-example) section for more info.

## **@SecondaryTables**

The [@SecondaryTables](http://docs.oracle.com/javaee/7/api/javax/persistence/SecondaryTables.html) annotation is used to group multiple [@SecondaryTable](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-secondarytable) annotations.

## **@SequenceGenerator**

The [@SequenceGenerator](http://docs.oracle.com/javaee/7/api/javax/persistence/SequenceGenerator.html) annotation is used to specify the database sequence used by the identifier generator of the currently annotated entity.  
See the [@SequenceGenerator mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#identifiers-generators-sequence-configured) section for more info.

## **@SqlResultSetMapping**

The [@SqlResultSetMapping](http://docs.oracle.com/javaee/7/api/javax/persistence/SqlResultSetMapping.html) annotation is used to specify the ResultSet mapping of a native SQL query or stored procedure.  
See the [SqlResultSetMapping mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#sql-composite-key-entity-associations_named-query-example) section for more info.

## **@SqlResultSetMappings**

The [@SqlResultSetMappings](http://docs.oracle.com/javaee/7/api/javax/persistence/SqlResultSetMappings.html) annotation is group multiple [@SqlResultSetMapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-sqlresultsetmapping) annotations.

## **@StoredProcedureParameter**

The [@StoredProcedureParameter](http://docs.oracle.com/javaee/7/api/javax/persistence/StoredProcedureParameter.html) annotation is used to specify a parameter of a [@NamedStoredProcedureQuery](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-namedstoredprocedurequery).  
See the [Using named queries to call stored procedures](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#sql-sp-named-query) section for more info.

## **@Table**

The [@Table](http://docs.oracle.com/javaee/7/api/javax/persistence/Table.html) annotation is used to specify the primary table of the currently annotated entity.  
See the [@Table mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#sql-custom-crud-secondary-table-example) section for more info.

## **@TableGenerator**

The [@TableGenerator](http://docs.oracle.com/javaee/7/api/javax/persistence/TableGenerator.html) annotation is used to specify the database table used by the identity generator of the currently annotated entity.  
See the [@TableGenerator mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#identifiers-generators-table-configured-mapping-example) section for more info.

## @Temporal

The [@Temporal](http://docs.oracle.com/javaee/7/api/javax/persistence/Temporal.html) annotation is used to specify the TemporalType of the currently annotated java.util.Date or java.util.Calendar entity attribute.  
See the [Basic temporal types](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#basic-datetime) chapter for more info.

## @Transient

The [@Transient](http://docs.oracle.com/javaee/7/api/javax/persistence/Transient.html) annotation is used to specify that a given entity attribute should not be persisted.  
See the [@Transient mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#events-jpa-callbacks-example) section for more info.

## @UniqueConstraint

The [@UniqueConstraint](http://docs.oracle.com/javaee/7/api/javax/persistence/UniqueConstraint.html) annotation is used to specify a unique constraint to be included by the automated schema generator for the primary or secondary table associated with the currently annotated entity.  
See the [Columns unique constraint](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#schema-generation-columns-unique-constraint) chapter for more info.

## @Version

The [@Version](http://docs.oracle.com/javaee/7/api/javax/persistence/Version.html) annotation is used to specify the version attribute used for optimistic locking.

## Hibernate Mapping Annotations

## @AccessType

The [@AccessType](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/AccessType.html) annotation is deprecated. You should use either the JPA [@Access](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-access) or the Hibernate native [@AttributeAccessor](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-attributeaccessor) annotation.

## @Any

The [@Any](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Any.html) annotation is used to define the any-to-one association which can point to one one of several entity types.  
See the [@Any mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#mapping-column-any) section for more info.

## @AnyMetaDef

The [@AnyMetaDef](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/AnyMetaDef.html) annotation is used to provide metadata about an @Any or @ManyToAny mapping.  
See the [@Any mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#mapping-column-any) section for more info.

## @AnyMetaDefs

The [@AnyMetaDefs](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/AnyMetaDefs.html) annotation is used to group multiple [@AnyMetaDef](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-anymetadef) annotations.

## @AttributeAccessor

The [@AttributeAccessor](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/AttributeAccessor.html) annotation is used to specify a custom [PropertyAccessStrategy](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/property/access/spi/PropertyAccessStrategy.html).  
Should only be used to name a custom PropertyAccessStrategy. For property/field access type, the JPA [@Access](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-access)annotation should be preferred.  
However, if this annotation is used with either value="property" or value="field", it will act just as the corresponding usage of the JPA [@Access](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-access) annotation.

## @BatchSize

The [@BatchSize](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/BatchSize.html) annotation is used to specify the size for batch loading the entries of a lazy collection.  
See the [Batch fetching](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#fetching-batch) section for more info.

## @Cache

The [@Cache](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Cache.html) annotation is used to specify the [CacheConcurrencyStrategy](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/CacheConcurrencyStrategy.html) of a root entity or a collection.  
See the [Caching](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#caching) chapter for more info.

## @Cascade

The [@Cascade](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Cascade.html) annotation is used to apply the Hibernate specific [CascadeType](http://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/CascadeType.html) strategies (e.g. CascadeType.LOCK, CascadeType.SAVE\_UPDATE, CascadeType.REPLICATE) on a given association.  
  
For JPA cascading, prefer using the [javax.persistence.CascadeType](http://docs.oracle.com/javaee/7/api/javax/persistence/CascadeType.html) instead.  
When combining both JPA and Hibernate CascadeType strategies, Hibernate will merge both sets of cascades.  
  
See the [Cascading](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#pc-cascade) chapter for more info.

## @Check

The [@Check](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Check.html) annotation is used to specify an arbitrary SQL CHECK constraint which can be defined at the class level.  
See the [Database-level checks](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#schema-generation-database-checks) chapter for more info.

## @CollectionId

The [@CollectionId](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/CollectionId.html) annotation is used to specify an identifier column for an idbag collection.  
You might want to use the JPA [@OrderColumn](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-ordercolumn) instead.

## @CollectionType

The [@CollectionType](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/CollectionType.html) annotation is used to specify a custom collection type.  
The collection can also name a [@Type](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-type), which defines the Hibernate Type of the collection elements.  
See the [Custom collection types](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#collections-custom) chapter for more info.

## @ColumnDefault

The [@ColumnDefault](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/ColumnDefault.html) annotation is used to specify the DEFAULT DDL value to apply when using the automated schema generator.  
The same behavior can be achieved using the definition attribute of the JPA [@Column](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-column) annotation.  
  
See the [Default value for a database column](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#schema-generation-column-default-value) chapter for more info.

## @Columns

The [@Columns](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Columns.html) annotation is used to group multiple JPA [@Column](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-column) annotations.  
  
See the [@Columns mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#mapping-column-read-and-write-composite-type-example) section for more info.

## @ColumnTransformer

The [@ColumnTransformer](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/ColumnTransformer.html) annotation is used to customize how a given column value is read from or write into the database.  
  
See the [@ColumnTransformer mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#mapping-column-read-and-write-example) section for more info.

## @ColumnTransformers

The [@ColumnTransformers](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/ColumnTransformers.html) annotation is used to group multiple [@ColumnTransformer](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-columntransformer) annotations.

## @CreationTimestamp

The [@CreationTimestamp](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/CreationTimestamp.html) annotation is used to specify that the currently annotated temporal type must be initialized with the current JVM timestamp value.  
  
See the [@CreationTimestamp mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#mapping-generated-CreationTimestamp) section for more info.

## @DiscriminatorFormula

The [@DiscriminatorFormula](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/DiscriminatorFormula.html) annotation is used to specify a Hibernate [@Formula](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-formula) to resolve the inheritance discriminator value.  
  
See the [@DiscriminatorFormula](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#entity-inheritance-single-table-discriminator-formula-example) section for more info.

## @DiscriminatorOptions

The [@DiscriminatorOptions](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/DiscriminatorOptions.html) annotation is used to provide the force and insert Discriminator properties.  
  
See the [Discriminator](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#entity-inheritance-discriminator) section for more info.

## @DynamicInsert

The [@DynamicInsert](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/DynamicInsert.html) annotation is used to specify that the INSERT SQL statement should be generated whenever an entity is to be persisted.  
  
By default, Hibernate uses a cached INSERT statement that sets all table columns. When the entity is annotated with the @DynamicInsert annotation, the PreparedStatement is going to include only the non-null columns.  
  
See the [@CreationTimestamp mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#mapping-generated-CreationTimestamp) section for more info on how @DynamicInsert works.

## @DynamicUpdate

The [@DynamicUpdate](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/DynamicUpdate.html) annotation is used to specify that the UPDATE SQL statement should be generated whenever an entity is modified.  
  
By default, Hibernate uses a cached UPDATE statement that sets all table columns. When the entity is annotated with the @DynamicUpdate annotation, the PreparedStatement is going to include only the columns whose values have been changed.  
  
See the [@DynamicUpdate](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#pc-managed-state-dynamic-update) section for more info.

## @Entity

The [@Entity](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Entity.html) annotation is deprecated. Use the JPA [@Entity](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-entity) annotation instead.

## @Fetch

The [@Fetch](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Fetch.html) annotation is used to specify the Hibernate specific [FetchMode](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/FetchMode.html) (e.g. JOIN, SELECT, SUBSELECT) used for the currently annotated association:  
  
See the [@Fetch mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#fetching-fetch-annotation) section for more info.

## @FetchProfile

The [@FetchProfile](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/FetchProfile.html) annotation is used to specify a custom fetching profile, similar to a JPA Entity Graph.  
  
[Fetch mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#fetching-strategies-dynamic-fetching-profile-example) section for more info.

## @FetchProfile.FetchOverride

The [@FetchProfile.FetchOverride](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/FetchProfile.FetchOverride.html) annotation is used in conjunction with the [@FetchProfile](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-fetchprofile) annotation, and it’s used for overriding the fetching strategy of a particular entity association.  
  
[Fetch profile](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#fetching-strategies-dynamic-fetching-profile-example) section for more info.

## @FetchProfiles

The [@FetchProfiles](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/FetchProfiles.html) annotation is used to group multiple [@FetchProfile](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-fetchprofile) annotations.

## @Filter

The [@Filter](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Filter.html) annotation is used to add filters to an entity or the target entity of a collection.  
  
See the [Filter mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#mapping-filter-example) section for more info.

## @FilterDef

The [@FilterDef](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/FilterDef.html) annotation is used to specify a @Filter definition (name, default condition and parameter types, if any).  
  
See the [Filter mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#mapping-filter-example) section for more info.

## @FilterDefs

The [@FilterDefs](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/FilterDefs.html) annotation is used to group multiple [@FilterDef](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-filterdef) annotations.

## @FilterJoinTable

The [@FilterJoinTable](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/FilterJoinTable.html) annotation is used to add @Filter capabilities to a join table collection.  
  
See the [FilterJoinTable mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#mapping-column-filter-join-table) section for more info.

## @FilterJoinTables

The [@FilterJoinTables](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/FilterJoinTables.html) annotation is used to group multiple [@FilterJoinTable](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-filterjointable) annotations.

## @Filters

The [@Filters](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Filters.html) annotation is used to group multiple [@Filter](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-filter) annotations.

## @ForeignKey

The [@ForeignKey](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/ForeignKey.html) annotation is deprecated. Use the JPA 2.1 [@ForeignKey](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-foreignkey) annotation instead.

## @Formula

The [@Formula](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Formula.html) annotation is used to specify an SQL fragment that is executed in order to populate a given entity attribute.  
  
See the [@Formula mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#mapping-column-formula-example) section for more info.

## @Generated

The [@Generated](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Generated.html) annotation is used to specify that the currently annotated entity attribute is generated by the database.  
  
See the [@Generated mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#mapping-generated-Generated) section for more info.

## @GeneratorType

The [@GeneratorType](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/GeneratorType.html) annotation is used to provide a [ValueGenerator](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/tuple/ValueGenerator.html) and a [GenerationTime](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/GenerationTime.html) for the currently annotated generated attribute.  
  
See the [@GeneratorType mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#mapping-generated-GeneratorType-example) section for more info.

## @GenericGenerator

The [@GenericGenerator](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/GenericGenerator.html) annotation can be used to configure any Hibernate identifier generator.  
  
See the [@GenericGenerator mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#identifiers-generators-GenericGenerator) section for more info.

## @GenericGenerators

The [@GenericGenerators](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/GenericGenerators.html) annotation is used to group multiple [@GenericGenerator](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-genericgenerator) annotations.  
@Immutable  
The [@Immutable](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Immutable.html) annotation is used to specify that the annotated entity, attribute, or collection is immutable.  
See the [@Immutable mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#entity-immutability) section for more info.

## @Index

The [@Index](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Index.html) annotation is deprecated. Use the JPA [@Index](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-index) annotation instead.

## @IndexColumn

The [@IndexColumn](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/IndexColumn.html) annotation is deprecated. Use the JPA [@OrderColumn](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-ordercolumn) annotation instead.

## @JoinColumnOrFormula

The [@JoinColumnOrFormula](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/JoinColumnOrFormula.html) annotation is used to specify that the entity association is resolved either through a FOREIGN KEY join (e.g. [@JoinColumn](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-joincolumn)) or using the result of a given SQL formula (e.g. [@JoinFormula](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-joinformula)).  
See the [@JoinColumnOrFormula mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#mapping-JoinColumnOrFormula) section for more info.

## @JoinColumnsOrFormulas

The [@JoinColumnsOrFormulas](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/JoinColumnsOrFormulas.html) annotation is used to group multiple [@JoinColumnOrFormula](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-joincolumnorformula) annotations.

## @JoinFormula

The [@JoinFormula](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/JoinFormula.html) annotation is used as a replacement for [@JoinColumn](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-joincolumn) when the association does not have a dedicated FOREIGN KEY column.  
See the [@JoinFormula mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#mapping-JoinFormula) section for more info.

## @LazyCollection

The [@LazyCollection](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/LazyCollection.html) annotation is used to specify the lazy fetching behavior of a given collection.  
  
The TRUE and FALSE values are deprecated since you should be using the JPA [FetchType](http://docs.oracle.com/javaee/7/api/javax/persistence/FetchType.html) attribute of the [@ElementCollection](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-elementcollection), [@OneToMany](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-onetomany), or [@ManyToMany](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-manytomany) collection.  
  
The EXTRA value has no equivalent in the JPA specification, and it’s used to avoid loading the entire collection even when the collection is accessed for the first time. Each element is fetched individually using a secondary query.  
  
See the [@LazyCollection mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#fetching-LazyCollection) section for more info.

## @LazyGroup

The [@LazyGroup](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/LazyGroup.html) annotation is used to specify that an entity attribute should be fetched along with all the other attributes belonging to the same group.  
  
To load entity attributes lazily, bytecode enhancement is needed. By default, all non-collection attributes are loaded in one group named "DEFAULT".  
  
This annotation allows defining different groups of attributes to be initialized together when access one attribute in the group.  
  
See the [@LazyGroup mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#BytecodeEnhancement-lazy-loading-example) section for more info.

## @LazyToOne

The [@LazyToOne](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/LazyToOne.html) annotation is used to specify the laziness options, represented by [LazyToOneOption](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/LazyToOneOption.html), available for a @OneToOne or @ManyToOne association.  
  
See the [@LazyToOne mapping example](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#associations-one-to-one-bidirectional-lazy) section for more info.

## @ListIndexBase

The [@ListIndexBase](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/ListIndexBase.html) annotation is used to specify the start value for a list index, as stored in the database.  
  
By default, List indexes are stored starting at zero. Generally used in conjunction with [@OrderColumn](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-ordercolumn).  
  
See the [@ListIndexBase mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#collections-customizing-ordered-list-ordinal) section for more info.

## @Loader

The [@Loader](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Loader.html) annotation is used to override the default SELECT query used for loading an entity loading.  
  
See the [Custom CRUD mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#sql-custom-crud-example) section for more info.

## @ManyToAny

The [@ManyToAny](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/ManyToAny.html) annotation is used to specify a many-to-one association when the target type is dynamically resolved.  
  
See the [@ManyToAny mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#mapping-column-many-to-any) section for more info.

## @MapKeyType

The [@MapKeyType](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/MapKeyType.html) annotation is used to specify the map key type.  
  
See the [@MapKeyType mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#collections-map-custom-key-type) section for more info.

## @MetaValue

The [@MetaValue](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/MetaValue.html) annotation is used by the [@AnyMetaDef](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-anymetadef) annotation to specify the association between a given discriminator value and an entity type.  
  
See the [@Any mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#mapping-column-any) section for more info.

## @NamedNativeQueries

The [@NamedNativeQueries](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/NamedNativeQueries.html) annotation is used to group multiple [@NamedNativeQuery](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-namednativequery) annotations.

@NamedNativeQuery

The [@NamedNativeQuery](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/NamedNativeQuery.html) annotation extends the JPA [@NamedNativeQuery](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-namednativequery) with Hibernate specific features.  
  
See the [Hibernate @NamedNativeQuery](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#sql-multiple-scalar-values-dto-NamedNativeQuery-hibernate-example) section for more info.

## @NamedQueries

The [@NamedQueries](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/NamedQueries.html) annotation is used to group multiple [@NamedQuery](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-namedquery) annotations.

## @NamedQuery

The [@NamedQuery](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/NamedQuery.html) annotation extends the JPA [@NamedQuery](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-namedquery) with Hibernate specific features.  
  
See the [@NamedQuery](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#jpql-api-hibernate-named-query-example) section for more info.

## @Nationalized

The [@Nationalized](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Nationalized.html) annotation is used to specify that the currently annotated attribute is a character type (e.g. String, Character, Clob) that is stored in a nationalized column type (NVARCHAR, NCHAR, NCLOB).  
  
See the [@Nationalized mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#basic-nationalized-example) section for more info.

## @NaturalId

The [@NaturalId](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/NaturalId.html) annotation is used to specify that the currently annotated attribute is part of the natural id of the entity.  
See the [Natural Ids](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#naturalid) section for more info.

## @NaturalIdCache

The [@NaturalIdCache](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/NaturalIdCache.html) annotation is used to specify that the natural id values associated with the annotated entity should be stored in the second-level cache.  
  
See the [@NaturalIdCache mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#naturalid-caching) section for more info.

## @NotFound

The [@NotFound](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/NotFound.html) annotation is used to specify the [NotFoundAction](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/NotFoundAction.html) strategy for when an element is not found in a given association.  
  
See the [@NotFound mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#associations-not-found) section for more info.

## @OnDelete

The [@OnDelete](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/OnDelete.html) annotation is used to specify the delete strategy employed by the currently annotated collection, array or joined subclasses. This annotation is used by the automated schema generation tool to generate the appropriate FOREIGN KEY DDL cascade directive.  
  
See the [@OnDelete cascade](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#pc-cascade-on-delete) chapter for more info.

## @OptimisticLock

The [@OptimisticLock](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/OptimisticLock.html) annotation is used to specify if the currently annotated attribute will trigger an entity version increment upon being modified.  
  
See the [Excluding attributes](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#locking-optimistic-exclude-attribute) section for more info.

## @OptimisticLocking

The [@OptimisticLocking](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/OptimisticLocking.html) annotation is used to specify the currently annotated an entity optimistic locking strategy.  
  
See the [Versionless optimistic locking](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#locking-optimistic-versionless) section for more info.

## @OrderBy

The [@OrderBy](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/OrderBy.html) annotation is used to specify a SQL ordering directive for sorting the currently annotated collection.  
  
It differs from the JPA [@OrderBy](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-orderby) annotation because the JPA annotation expects a JPQL order-by fragment, not an SQL directive.  
  
See the [@OrderBy mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#collections-customizing-ordered-by-sql-clause) section for more info.

## @ParamDef

The [@ParamDef](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/ParamDef.html) annotation is used in conjunction with [@FilterDef](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-filterdef) so that the Hibernate Filter can be customized with runtime-provided parameter values.  
  
See the [Filter mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#mapping-filter-example) section for more info.

## @Parameter

The [@Parameter](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Parameter.html) annotation is a generic parameter (basically a key/value combination) used to parametrize other annotations, like [@CollectionType](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-collectiontype), [@GenericGenerator](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-genericgenerator), and [@Type](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-type), [@TypeDef](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-typedef).

## @Parent

The [@Parent](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Parent.html) annotation is used to specify that the currently annotated embeddable attribute references back the owning entity.  
  
See the [@Parent mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#mapping-Parent) section for more info.

## @Persister

The [@Persister](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Persister.html) annotation is used to specify a custom entity or collection persister.  
  
For entities, the custom persister must implement the [EntityPersister](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/persister/entity/EntityPersister.html) interface.  
  
For collections, the custom persister must implement the [CollectionPersister](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/persister/collection/CollectionPersister.html) interface.  
  
See the [@Persister mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#entity-persister) section for more info.

## @Polymorphism

The [@Polymorphism](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Polymorphism.html) annotation is used to define the [PolymorphismType](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/PolymorphismType.html) Hibernate will apply to entity hierarchies.  
  
See the [@Polymorphism](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#entity-inheritance-polymorphism) section for more info.

## @Proxy

The [@Proxy](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Proxy.html) annotation is used to specify a custom proxy implementation for the currently annotated entity.  
  
See the [@Proxy mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#entity-proxy) section for more info.

## @RowId

The [@RowId](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/RowId.html) annotation is used to specify the database column used as a ROWID pseudocolumn. For instance, Oracle defines the [ROWID pseudocolumn](https://docs.oracle.com/cd/B19306_01/server.102/b14200/pseudocolumns008.htm) which provides the address of every table row.  
  
See the [@RowId mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#identifiers-rowid) section for more info.

## @SelectBeforeUpdate

The [@SelectBeforeUpdate](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/SelectBeforeUpdate.html) annotation is used to specify that the currently annotated entity state be selected from the database when determining whether to perform an update when the detached entity is reattached.  
  
See the [OptimisticLockType.DIRTY mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#locking-optimistic-lock-type-dirty-example) section for more info on how @SelectBeforeUpdate works.

## @Sort

The [@Sort](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Sort.html) annotation is deprecated. Use the Hibernate specific [@SortComparator](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-sortcomparator) or [@SortNatural](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-sortnatural) annotations instead.

## @SortComparator

The [@SortComparator](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/SortComparator.html) annotation is used to specify a Comparator for sorting the Set/Map in-memory.  
  
See the [@SortComparator mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#collections-unidirectional-sorted-set-custom-comparator-example) section for more info.

## @SortNatural

The [@SortNatural](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/SortNatural.html) annotation is used to specify that the Set/Map should be sorted using natural sorting.  
  
See the [@SortNatural mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#collections-bidirectional-sorted-set-example) section for more info.

## @Source

The [@Source](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Source.html) annotation is used in conjunction with a @Version timestamp entity attribute indicating the [SourceType](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/SourceType.html)of the timestamp value.  
  
See the [Database-generated version timestamp mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#locking-optimistic-version-timestamp-source-mapping-example) section for more info.

## @SQLDelete

The [@SQLDelete](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/SQLDelete.html) annotation is used to specify a custom SQL DELETE statement for the currently annotated entity or collection.  
  
See the [Custom CRUD mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#sql-custom-crud-example) section for more info.

## @SQLDeleteAll

The [@SQLDeleteAll](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/SQLDeleteAll.html) annotation is used to specify a custom SQL DELETE statement when removing all elements of the currently annotated collection.  
  
See the [Custom CRUD mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#sql-custom-crud-example) section for more info.

## @SqlFragmentAlias

The [@SqlFragmentAlias](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/SqlFragmentAlias.html) annotation is used to specify an alias for a Hibernate [@Filter](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-filter).  
  
The alias (e.g. myAlias) can then be used in the @Filter condition clause using the {alias} (e.g. {myAlias}) placeholder.  
  
See the [@SqlFragmentAlias mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#mapping-column-filter-sql-fragment-alias) section for more info.

## @SQLInsert

The [@SQLInsert](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/SQLInsert.html) annotation is used to specify a custom SQL INSERT statement for the currently annotated entity or collection.  
  
See the [Custom CRUD mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#sql-custom-crud-example) section for more info.

## @SQLUpdate

The [@SQLUpdate](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/SQLUpdate.html) annotation is used to specify a custom SQL UPDATE statement for the currently annotated entity or collection.  
  
See the [Custom CRUD mapping](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#sql-custom-crud-example) section for more info.

## **@Subselect**

The [@Subselect](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Subselect.html) annotation is used to specify an immutable and read-only entity using a custom SQL SELECT statement.  
  
See the [Mapping the entity to a SQL query](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#entity-sql-query-mapping) section for more info.

## **@Synchronize**

The [@Synchronize](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Synchronize.html) annotation is usually used in conjunction with the [@Subselect](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-subselect) annotation to specify the list of database tables used by the @Subselect SQL query.  
  
See the [Mapping the entity to a SQL query](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#entity-sql-query-mapping) section for more info.

## **@Table**

The [@Table](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Table.html) annotation is used to specify additional information to a JPA [@Table](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-table) annotation, like custom INSERT, UPDATE or DELETE statements or a specific [FetchMode](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/FetchMode.html).

## **@Tables**

The [@Tables](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Tables.html) annotation is used to group multiple [@Table](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-table) annotations.

## **@Target**

The [@Target](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Target.html) annotation is used to specify an explicit target implementation when the currently annotated association is using an interface type.

## **@Tuplizer**

The [@Tuplizer](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Tuplizer.html) annotation is used to specify a custom tuplizer for the currently annotated entity or embeddable.

## **@Tuplizers**

The [@Tuplizers](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Tuplizers.html) annotation is used to group multiple [@Tuplizer](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-tuplizer) annotations.

## **@Type**

The [@Type](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Type.html) annotation is used to specify the Hibernate [@Type](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/type/Type.html) used by the currently annotated basic attribute.

## **@TypeDef**

The [@TypeDef](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/TypeDef.html) annotation is used to specify a [@Type](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/type/Type.html) definition which can later be reused for multiple basic attribute mappings.

## **@TypeDefs**

The [@TypeDefs](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/TypeDefs.html) annotation is used to group multiple [@TypeDef](http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-hibernate-typedef) annotations.

## **@UpdateTimestamp**

The [@UpdateTimestamp](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/UpdateTimestamp.html) annotation is used to specify that the currently annotated timestamp attribute should be updated with the current JVM timestamp whenever the owning entity gets modified.

## **@ValueGenerationType**

The [@ValueGenerationType](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/ValueGenerationType.html) annotation is used to specify that the current annotation type should be used as a generator annotation type.

## **@Where**

The [@Where](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/Where.html) annotation is used to specify a custom SQL WHERE clause used when fetching an entity or a collection.

## **@WhereJoinTable**

The [@WhereJoinTable](https://docs.jboss.org/hibernate/orm/5.3/javadocs/org/hibernate/annotations/WhereJoinTable.html) annotation is used to specify a custom SQL WHERE clause used when fetching a join collection table.