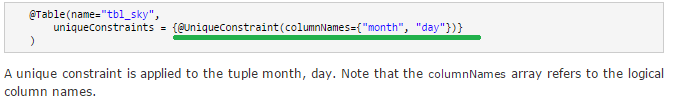
Notes is Prepared from Java Docs

1. 
2. **@Version**
3. 

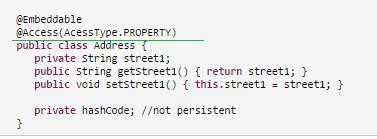
**2.2.2.2. Access type**

**@Access**

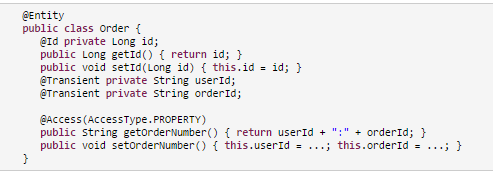
1. By default the access type of a class hierarchy is defined by the position of the @Id or @EmbeddedId annotations. If these annotations are on a field, then only fields are considered for persistence and the state is accessed via the field. If there annotations are on a getter, then only the getters are considered for persistence and the state is accessed via the getter/setter.

Now let’s see the example for access type in different scenario

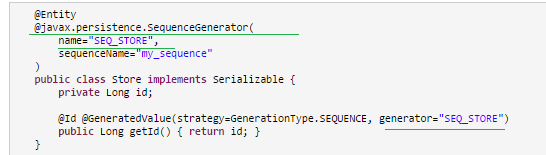
1. Scenario 1🡺 To force the access type on a given class, use the @Access annotation as showed below:



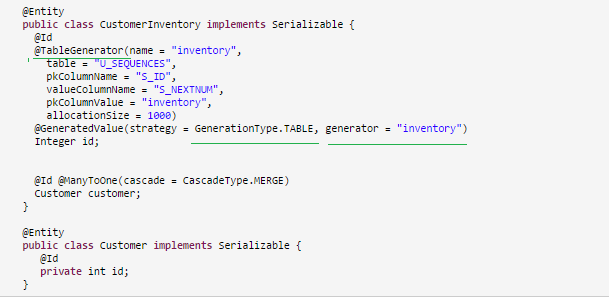
1. Scenario 2🡺 you can also override the access type of a single property while keeping the other properties standard🡺 **In this example, the default access type is FIELD except for the orderNumber property. Note that the corresponding field, if any must be marked as @Transient or transient.**



1. @javax.persistence.SequenceGenerator



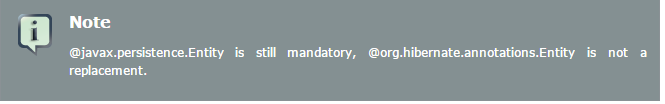
1. @Secondary🡺 used if one entity is mapped to more than one Table🡺 Needed to see the example
2. @MapsId
3. @IdClass
4. @TableGenerator



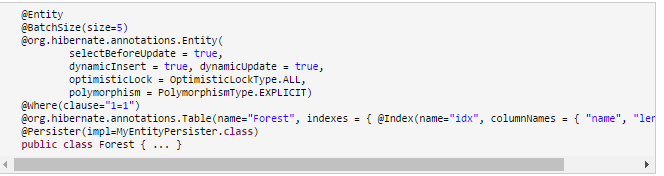
1. @PrimaryKeyJoinColumn
2. @OrderBy
3. @OrderColumn
4. @MapKeyColumn🡺
5. @MapKeyClass 🡺You can also use @MapKeyClass to define the type of the key if you don't use generics (at this stage, you should wonder why at this day and age you don't use generics).
6. @SqlResultSetMapping

##### Indexed collections

1. List and Map is called as indexed collections



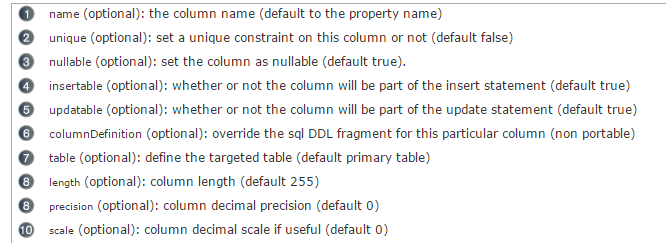
1. Here are some additional Hibernate annotation extensions
2. @org.hibernate.annotations.BatchSize
3. @org.hibernate.annotations.Proxy
4. @org.hibernate.annotations.Where
5. @org.hibernate.annotations.Where
6. @org.hibernate.annotations.Check
7. @OnDelete(action=OnDeleteAction.CASCADE)
8. @Immutable
9. @Persister
10. @OnDelete



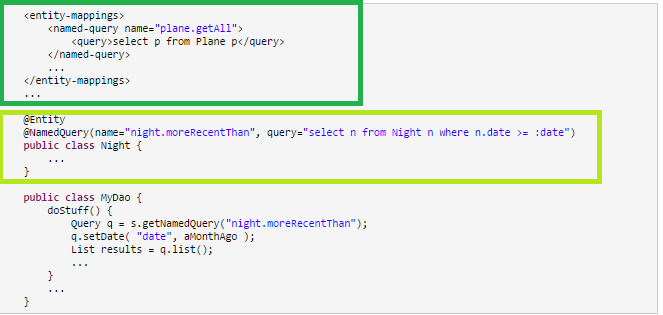
Doubts from JavaDocs

1. Needed to know the purpose of <jta-data-source> in the persistence.xml file
2. Needed to know the purpose of transaction-type =”Resource-local” attribute in the <Persistence-unit> element
3. Basically needed to know why JPA uses EntityManager and Hibernate uses Session
4. Revise List of JPA keywords
5. Constructor Expressions🡺 Needed to see the Example
6. Needed Clear understanding on Case used in JPA
7. 

Following are the list of the attributes that can be used in the @Column annotation



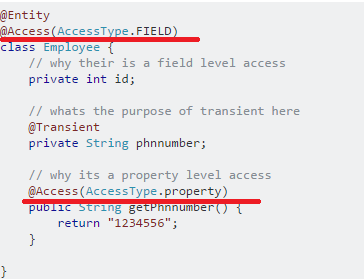
1. a class can have only one @Id annotation or it can have multiple @ID annotation
2. **Optimistic Locking in Hibernate**
3. Following diagram shows the @ based and it’s equivalent XML based Named Queries in Hibernate



Explanation

# [What is the purpose of AccessType.FIELD, AccessType.PROPERTY and @Access](http://stackoverflow.com/questions/13874528/what-is-the-purpose-of-accesstype-field-accesstype-property-and-access)

1. By default the access type is defined by the place where you put your identifier annotation (@Id).
2. If you put it on the field - it will be AccessType.FIELD, if you put it on the getter - it will be AccessType.PROPERTY.
3. Sometimes you might want to annotate not fields but properties (e.g. because you want to have some arbitrary logic in the getter or because you prefer it that way.) In such situation you must define a getter and annotate it as AccessType.PROPERTY.
4. **As far as I remember, if you specify either AccessType.FIELD or AccessType.PROPERTY on any of the entity fields / methods you must specify the default behavior for the whole class. And that's why you need to have AccessType.FIELD on the class level (despite that AccessType.FIELD is the default value.)**



1. @UniqueConstraint
2. 
3. 