JPA

Java Persistence API is a collection of classes and methods to persistently store the vast amounts of data into a database.

@**Entity** - A class which should be persisted in a database. JPA uses a database table for every entity. Persisted instances of the class will be represented as one row in the table.

@**Cacheable** – ALL, NONE, ENABLE\_SELECTIVE, DISABLE\_SELECTIVE  
 cache.contains(Animal.class, idAnimal)  
<http://www.thejavageek.com/2014/09/25/jpa-caching-example/>

@**MappedSuperclass** – Designates a class whose mapping information is applied to the entities that inherit from it. A mapped superclass has no separate table defined for it.

no table exists for the mapped superclass itself, mapping apply to subclasses. Mapping can be overridden with ***AttributeOverride and AssociationOverride***

@**ID** - Specifies the primary key of an entity.

@**Version** – version of an Entity class, ensures integrity on merging  
only on 1 property/field of a class, must be mapped to corresponding table.

@**Transient** – field will not be saved in the DB, it is not persistent. Used for mapping a entity class, mapped superclass, or embeddable class.

@**EnumType** – Defines mapping for enumerated types.

@**Temporal –** for persistent fields: Date & Calendar

@**OrderBy** – when the association or collection is retrieved.  
 orderby\_list::= orderby\_item [,orderby\_item]\*  
 orderby\_item::= [property\_or\_field\_name] [ASC | DESC]  
if not specified = by the primary key

@**MapKey** – for java.utils.Map when map key is itself the PK or a persistent field or property of the entity that is the value of the map.  
The *MapKeyClass* annotation is not used when MapKey is specified and vice versa.

**CascadeType** – only for parent-child associations = Parent entity state transition being cascaded to its Child entities.  
All – Toate restu.  
 @OneToOne(mappedBy = "post",  
        cascade = CascadeType.ALL, orphanRemoval = true)  
 private PostDetails details;  
si pe @OneToMany !

1. **CascadeType.PERSIST** : cascade type presist means that save() or persist() operations cascade to related entities.
2. **CascadeType.MERGE** : cascade type merge means that related entities are merged when the owning entity is merged. (Update)
3. **CascadeType.REFRESH** : cascade type refresh does the same thing for the refresh() operation. (PULL)
4. **CascadeType.REMOVE** : cascade type remove removes all related entities association with this setting when the owning entity is deleted.
5. **CascadeType.DETACH** : cascade type detach detaches all related entities if a “manual detach” occurs.
6. **CascadeType.ALL** : cascade type all is shorthand for all of the above cascade operations.

**FetchType - EAGER / LAZY**

Cu LAZY – se face load doar la datele din entitaea parinte.  
Cu EAGER se face load si la entitatile fiu. Ex. Lista studenti -> universitate.

LAZY = fetch when needed  
EAGER = fetch immediately@Relatie(fetch = FetchType.LAZY)

FetchType valueOf(String name)  
Returns the enum constant of this type with the specified name.

FetchType[] values()  
Returns an array containing the constants of this enum type.

**@GeneratedValue**(strategy = ***GenerationType***.IDENTITY)

*GenerationType*: Defines the types of primary key generation strategies.  
*GenerationType* **AUTO**:   
Indicates that the persistence provider should pick an appropriate strategy for the particular database. If table has defined any default value or it has defined any auto increment in table.  
*GenerationType* **IDENTITY**Primary keys for the entity will be chosen using DB identity column. allows an integer/bigint column to be auto-incremented on demand.  
*GenerationType* **SEQUENCE**  
Indicates that the persistence provider must assign primary keys for the entity using a database sequence. We need to create a sequence generator in database and refer that name in the code.

@GeneratedValue(strategy=GenerationType.SEQUENCE, generator="course\_seq")

@SequenceGenerator(

name="course\_seq",

sequenceName="course\_sequence",

allocationSize=20

)

*GenerationType* **TABLE**  
Indicates that the persistence provider must assign primary keys for the entity using an underlying database table to ensure uniqueness.  
=SEQUENCE pe baza unui table.