Doctrine 2 Sheets

Classes annotations / Main

@Entity
repositoryClass

@HasLifecycleCallbacks

@Table
name
indexes

Properties annotations / Main

@Column

type	string, integer, smallint, bigint, boolean, decimal, date, time, datetime, text, object, array, float
name	column name *(string)*
length	column length *(integer)*
unique	unique key *(true or false)*
nullable	column can be null *(true or false)*

@Id

@Index

name	index name *(string)*
columns	related columns *(strings array)*

@GeneratedValue *with @Id*

strategy	AUTO, SEQUENCE, TABLE, IDENTITY, NONE

Properties annotations / Associations

@OneToOne

targetEntity	FQCN of the referenced target entity *(string)*
inversedBy	field in the entity that is the inverse side *(string)*
cascade	*(persist, remove, merge, detach, all)*
fetch	fetch type *(LAZY or EAGER)*
orphanRemoval	remove orphan *(true or false)*

@OneToMany

targetEntity	FQCN of the target entity *(string)*
cascade	*(persist, remove, merge, detach, all)*
orphanRemoval	remove orphan *(true or false)*
mappedBy	property on the targetEntity that is the owning side *(string)*

@ManyToOne

targetEntity	FQCN of the target entity *(string)*
cascade	*(persist remove, merge, detach, all)*
fetch	fetch type *(LAZY or EAGER)*

@ManyToMany

targetEntity	FQCN of the target entity *(string)*
mappedBy	property on the targetEntity that is the owning side *(string)*
inversedBy	field in the entity that is the inverse side *(string)*
cascade	*(persist, remove, merge, detach, all)*
fetch	fetch type *(LAZY or EAGER)*

@JoinTable *with @OneToMany or @ManyToMany*

name	Database name of the join-table
joinColumns	An array of @JoinColumn
inverseJoinColumns	An array of @JoinColumn

@JoinColumn *with @ManyToOne or @OneToOne*

name	Column name that holds the foreign key
referencedColumnName	Name of the primary key identifier used for join
unique	is this relation exclusive between the affected entities
nullable	related entity is required
onDelete	Cascade Action (Database-level)
onUpdate	Cascade Action (Database-level)

@OrderBy

Class annotations / Inheritance

@DiscriminatorColumn

@DiscriminatorMap

@InheritanceType

@MapperSuperclass

Method annotations / Callbacks

require @HasLifecycleCallbacks

@PostLoad, @PostPersist, @PostRemove, @PostUpdate, @PrePersist, @PreRemove, @PreUpdate

Associations Example / One-To-One Bidirectional

```php
<?php
    /** @Entity */
    class Customer
    {
        /**
         * @OneToOne(targetEntity="Cart", mappedBy="customer")
         */
        private $cart;
    }

    /** @Entity */
    class Cart
    {
        /**
         * @OneToOne(targetEntity="Customer", inversedBy="cart")
         * @JoinColumn(name="customer_id", referencedColumnName="id")
         */
        private $customer;
    }
```

Associations Example / One-To-Many Bidirectional

```php
<?php
 /** @Entity */
 class Product
 {
    /**
    * @OneToMany(targetEntity="Feature", mappedBy="product")
    */
    private $features;

    public function __construct() {
       $this->features = new \Doctrine\Common\Collections\ArrayCollection();
    }
 }

 /** @Entity */
 class Feature
 {
    /**
    * @ManyToOne(targetEntity="Product", inversedBy="features")
    * @JoinColumn(name="product_id", referencedColumnName="id")
    */
    private $product;
 }
```

Associations Example / One-To-Many Self Referencing

```php
<?php
 /** @Entity */
 class Category
 {
    /**
    * @OneToMany(targetEntity="Category", mappedBy="parent")
    */
    private $children;

    /**
    * @ManyToOne(targetEntity="Category", inversedBy="children")
    * @JoinColumn(name="parent_id", referencedColumnName="id")
    */
    private $parent;

    public function __construct() {
       $this->children = new \Doctrine\Common\Collections\ArrayCollection();
    }
 }
```

Associations Example / Many-To-Many Bidir - Ordered

```php
<?php
 /** @Entity */
 class User
 {
    /**
    * @ManyToMany(targetEntity="Group", inversedBy="users")
    * @OrderBy({"name" = "ASC"})
    * @JoinTable(name="users_groups",
    *     joinColumns={@JoinColumn(name="user_id", referencedColumnName="id")},
    *     inverseJoinColumns={@JoinColumn(name="group_id", referencedColumnName="id")}
    * )
    */
    private $groups;

    public function __construct() {
       $this->groups = new \Doctrine\Common\Collections\ArrayCollection();
    }
 }

 /** @Entity */
 class Group
 {
    /**
    * @ManyToMany(targetEntity="User", mappedBy="groups")
    */
    private $users;

    public function __construct() {
       $this->users = new \Doctrine\Common\Collections\ArrayCollection();
    }
 }
```

Inheritance Example

```php
<?php
 namespace MyProject\Model;
 /**
  * @Entity
  * @InheritanceType("JOINED")
  * @DiscriminatorColumn(name="discr", type="string")
  * @DiscriminatorMap({"person" = "Person", "employee" = "Employee"})
  */
 class Person
 {
 }

 /** @Entity */
 class Employee extends Person
 {
 }
```