**MUCHOS A UNO – UNIDIRECCIONAL – 01**

**Pro JPA 2 in Java EE 8 - Página 133**

|  |  |
| --- | --- |
| **SOURCE** | **TARGET** |
|  | |
| * **Employee** is the many side * And the source of the relationship | * **Department** is the one side * And the target of the relationship |
|  | |
| * Employee has an attribute called **department** that will contain a reference to a single Department instance * The attribute is not showing in the Employee class but is implied by the presence of the relationship. |  |
|  | * Solo dos tags: @Entity y @Id * No se le agrega nada mas |
| * A many-to-one mapping is defined by annotating the attribute (the attribute that refers to the target entity - **department**) in the source entity with the **@ManyToOne** annotation. |  |
|  | |
| * The **EMPLOYEE** table has a ***foreign key*** column named **DEPT\_ID** that references the **DEPARTMENT** TABLE. * **DEPT\_ID** is the ***join column*** that associates the Employee and Department entities. |  |
|  |  |
| * Independent of source and target sides, one of the two sides will have the join column in its table. * That side is called the **owning side** or the **owner** of the relationship. * The absence of the **mappedBy** element in the mapping annotation implies **ownership** of the relationship. * **Logical mapping** (**@ManyToOne**): By convention, it should appear first. * Followed by the **physical mapping** (**@JoinColumn**) | * The side that does not have the join column is called the **non-owning** or **inverse side**. * While the presence of the **mappedBy** element means the entity is on the **inverse side** of the relationship. * The **mappedyBy** is used when the relationship is **bidirectional**. |