

# Primary Key Tanımları

# Primary Key

- Bir kaydı tablo genelinde benzersiz biçimde tanımlayan **sütun veya sütunlardır**
- Entity sınıf içerisinde @Id anotasyonu, **identifier** property yani **primary key** tanımlar
- İki farklı PK oluşturma yöntemi mevcuttur
  - Assigned PK yöntemi
  - Sentetik (Surrogate) PK yöntemi

# Assigned Primary Key

- Programcı tarafından set edilen identifier “**assigned identifier**” olarak adlandırılır
- Programcı tarafından atanan identifier değerleri **genellikle iş mantığı ile ilgili** olabilmektedir
- İş mantığı açısından da bir anlam ifade eden bu değerlere **natural key** adı verilir
- Identifier alanına değer set edilmeden yapılan persist işlemleri **başarısız** olacaktır

# Assigned Primary Key

```
@Entity
@Table(name="T_PERSON")
public class Person {

    @Id
    private String ssNo;

    public String getSsNo() {
        return ssNo;
    }

    public void setSsNo(String ssNo) {
        this.ssNo = ssNo;
    }

    ...
}
```

→ ID değeri yeni bir entity yaratılırken uygulama tarafında set edildikten sonra persist işlemi gerçekleştirilmelidir

# Surrogate (Sentetik) Primary Key

- Veritabanı veya uygulama tarafından **otomatik olarak üretilen** benzersiz bir değerdir
- Son kullanıcı için hiçbir anlam ifade etmez
- İş mantığı ile **alakalı değildir**
- **Sentetik identifier** olarak da bilinir

# Surrogate (Sentetik) Primary Key

```
@Entity
public class Pet {

    @Id
    @GeneratedValue
    private Long id;

    private String name;

    ...
}
```

→ **@GeneratedValue** ile ID değerinin otomatik üretilmesi sağlanır

Değişik **id üretme yöntemleri** vardır  
GeneratedValue annotasyonu içinde **strategy** attribute değeri ile belirlenir

- AUTO (default)
- IDENTITY
- SEQUENCE
- TABLE

# Sentetik ID Üretme Yöntemleri

```
@Entity
public class Owner {
    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    private Long id;
    ...
}
```

Default stratejidir. DB'nin ID üretme stratejisi tercih edilir

```
@Entity
public class Vet {
    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private Long id;
    ...
}
```

ID değeri kayıt DB'ye insert edilirken DB tarafından Oluşturulur. Mevcut kayıtlara göre max PK değeri elde edilir

# Sentetik ID Üretme Yöntemleri

```
@Entity
public class Pet {
    @Id
    @GeneratedValue(strategy=GenerationType.SEQUENCE)
    private Long id;
    ...
}
```

DB genelinde tek bir  
sequence kullanılır  
(HIBERNATE\_SEQUENCE)

```
@Entity
public class Pet {
    @Id
    @GeneratedValue(generator = "pet_seq_gen")
    @SequenceGenerator(name = "pet_seq_gen", sequenceName = "PET_SEQ",
        initialValue = 10, allocationSize = 100)
    private Long id;
    ...
}
```

SequenceGenerator tanımlayarak her tablo için  
ayrı sequence'ların kullanılması da mümkündür



# Primary Key ve Property Access İlişkisi

- Bir kaydı tablo genelinde benzersiz biçimde tanımlayan **sütun veya sütunlardır**
- Entity sınıf içerisinde @Id anotasyonu, **identifier** property yani **primary key** tanımlar
- İki farklı PK oluşturma yöntemi mevcuttur
  - Assigned PK yöntemi
  - Sentetik (Surrogate) PK yöntemi
- @Id'nin kullanıldığı yer **persistent alanlara erişim yöntemini** belirler
  - Eğer field üzerinde kullanılmış ise field düzeyinde erişim

# Primary Key – Field Level Access

```
@Entity
@Table(name="T_PET")
public class Pet {

    @Id
    private Long id;

    @Column(name="PET_NAME")
    private String name;

    ...
}
```

→ @Id annotasyonu field üzerinde kullanılmış ise entity'nin property değerlerinin set edilmesi field level access ile yapılır

Bu durumda entity sınıf içerisinde getter/setter metotlarına ihtiyaç yoktur

Setter/getter metotları tamamen iş mantığına göre, ihtiyaç varsa tanımlanırlar

# Primary Key – Getter Level Access

@Entity

```
public class Pet {
```

```
    private Long id;  
    private String name;
```

```
@Id  
    public Long getId() {  
        return id;  
    }
```

```
    public void setId(Long id) {  
        this.id = id;  
    }
```

```
@Column(name="PET_NAME")  
    public String getName() {  
        return name;  
    }  
    public void setName(String name) {  
        this.name = name;  
    }  
}
```

→ @Id annotasyonu getId() metodunda tanımlanırsa sınıfın propertylerine erişim, property değerlerinin yüklenmesi getter/setter metotları üzerinden yürütülür

Field düzeyinde lazy load dezavantajı dışında field level access, getter level access'e tercih edilmelidir

# İletişim

- Harezmi Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- <http://www.java-egitimleri.com>
- [info@java-egitimleri.com](mailto:info@java-egitimleri.com)

