Primary Key Tanımlama Yöntemleri

harezmi bilişim çözümleri

Her Tablo için Ayrı Bir Sequence Tanımlama Yöntemi

```
@MappedSuperclass
public abstract class BaseEntity {
    @Id

    @GeneratedValue(strategy = GenerationType.SEQUENCE,
                    generator = "sequenceStyleGenerator")

    @org.hibernate.annotations.GenericGenerator(
        name = "sequenceStyleGenerator",
        strategy = "sequence",
        parameters = @org.hibernate.annotations.Parameter(
                    name = "prefer_sequence_per_entity",
                    value = "true"))
    private Long id;
    ...
}
```

Hibernate'in SequenceStyleGenerator isimli sınıfının bir parametresidir. Bu sayede şema genelinde tek bir sequence tanımlamak yerine her bir entity tablosu için ayrı bir sequence üretir. Sequence isimleri default olarak TABLOADI_SEQ şeklinde olacaktır.

Sentetik ID Üretme Yöntemleri: UUID

- UUID, 128 bit benzersiz bir ifadedir
- Canonical gösterimi 8-4-4-4-12 şeklinde 36 karakterlik String değer şeklindedir
  - Örnek; 123e4567-e89b-12d3-a456-426655440000

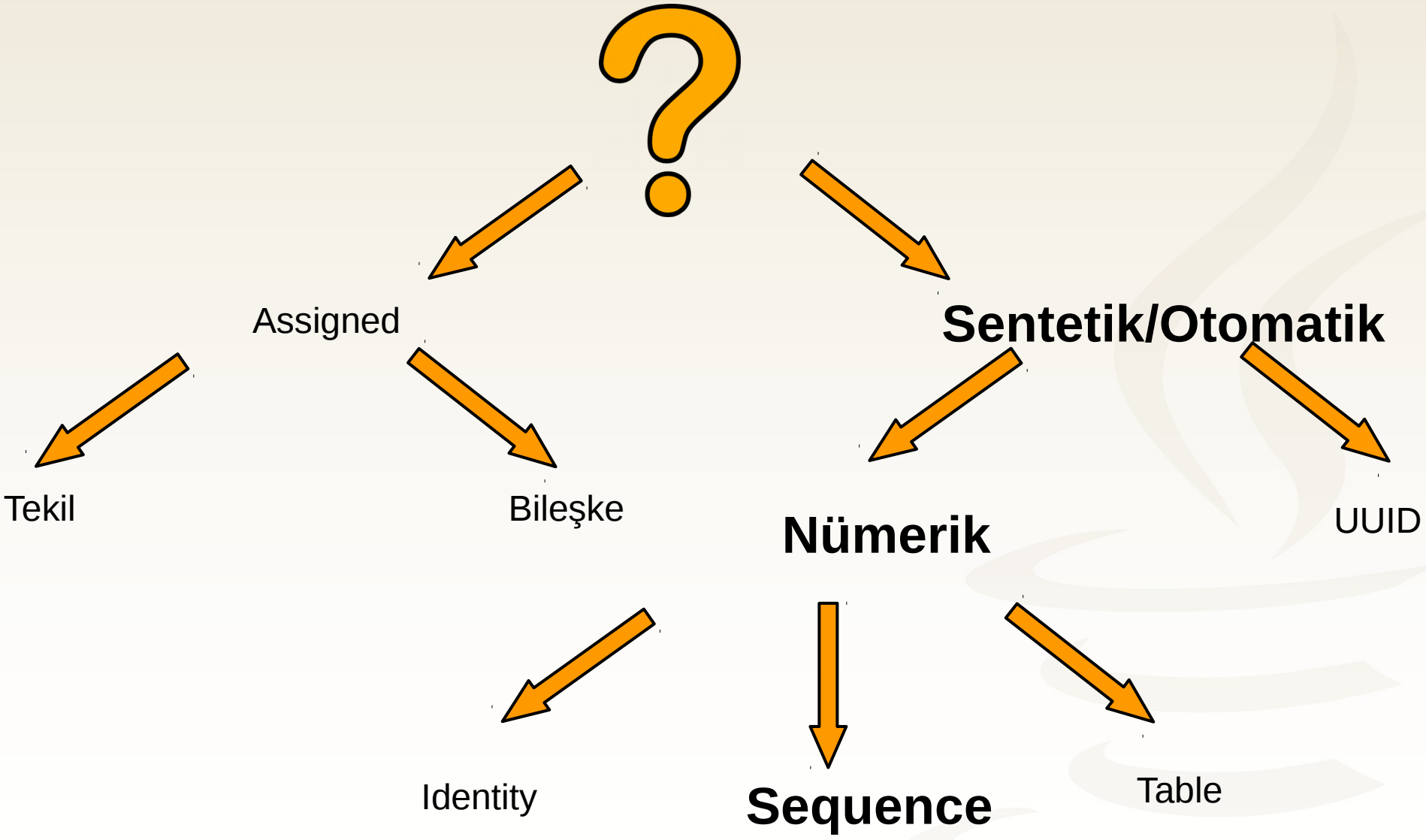
```
@Entity
public class Visit {
    @Id
    @GeneratedValue(generator="uuid_gen")
    @org.hibernate.annotations.GenericGenerator(
                                    name="uuid_gen",strategy="uuid")
    private String id;
    ...
}
```

Sentetik ID Üretme Yöntemleri: Tablo Üzerinden Sequence

- Veritabanı sequence kabiliyetini desteklemiyorsa kullanılabilir
- PK değerleri ayrı bir tabloda yönetilir
- Bir sonraki PK değerini elde etmek için TX içerisinde kayıt düzeyinde lock kullanılır
- Maliyetli bir yöntemdir

```
@Entity
public class PetType {
    @Id
    @GeneratedValue(strategy = GenerationType.TABLE, generator = "petTypeGen")
    @TableGenerator(name = "petTypeGen", table = "ID_GEN", pkColumnName =
"GEN_KEY", valueColumnName = "GEN_VALUE", pkColumnValue = "PET_TYPE_ID")
    private Long id;
    ...
}
```

Hangi PK Yöntemi Tercih Edilmeli?



Hangi PK Yöntemi Tercih Edilmeli?

- Eğer **naturel PK tekil ve nümerik bir değer ise** performans açısından sentetik PK yöntemi ile hemen hemen aynıdır
- Ancak iş mantığından bağımsız **sentetik bir PK daha çok esneklik** sağlamaktadır
- Web uygulamalarında veriye erişim, auditing, kayıt düzeyinde yetkilendirme gibi **işlemler daha kolay/standart biçimde** gerçekleştirilebilir

Hangi PK Yöntemi Tercih Edilmeli?

- **Bileşke PK değerleri** ayrıca veriye erişim açısından join'lerde ve indekslemede performans ve veri depolama alanı açısından daha verimsizdir
- Mecbur kalınmadıkça **tercih edilmeme**lidir
- **UUID** stratejisi daha çok cluster ortamlar için benzersiz PK değeri oluşturmak için tercih edilir
- Ancak **veri depolama alanı** ve **indeksleme** noktalarından dezavantaj yaratabilir

Hangi PK Yöntemi Tercih Edilmeli?

- **Identity** ancak sequence tercih edilemiyorsa kullanılmalıdır, JDBC **batch insert'leri desteklemez**
- **Tablo** üzerinden PK yönetimi **en az verimli** yöntemdir
- PK'nın elde edilmesi için **ayrı bir TX**'e gerek duyar
- Ayrıca **row-lock** ölçeklenmede problemler yaratabilmektedir

Hangi PK Yöntemi Tercih Edilmeli?

JAVA Eğitimleri

- **En uygun yöntem Sequence** olarak karşımıza çıkmaktadır
- Özellikle **pooled** veya **pooled-lo** yöntemi kullanan bir **sequence stratejisi** oldukça verimlidir
  - Pooled/pooled-lo yöntemlerinin devreye girmesi için *hibernate.id.new_generator_mappings=true* tanımlı olması gerekir
  - Bu tanımla default olarak pooled aktive olur
  - *hibernate.id.optimizer.prefer_lo=true* tanımı ile pooled-lo'ya geçiş yapılabilir

Veritabanlarının Sentetik PK Üretme Yöntemleri

Veritabanı	Sentetik PK Yöntemi
Oracle	Sequence,Identity (Oracle12c)
MSSQL	Identity,Sequence(MS SQL2012)
PostgreSQL	Sequence,Serial Type
MySQL	Auto_Increment
DB2	Identity,Sequence
HSQLDB/H2	Identity,Sequence

İletişim

- Harezmi Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- http://www.java-egitimleri.com
- info@java-egitimleri.com