

Date ve Time Tipleri



Date ve Time Tipleri

- Hibernate, default olarak **java.util.Date** property'sini **"timestamp"** ile eşler
- **Date** veya **Calendar** tiplerin SQL tipi **timestamp** dışında **time** veya **date** yapılmak istenirse bunun için **@Temporal** annotasyonu kullanılır

```
@Entity
public class Pet {
```

```
    @Temporal(TemporalType.DATE)
    @Column(name="BIRTH_DATE")
    private java.util.Date birthDate;
```

```
}
```

TemporalType.TIMESTAMP
TemporalType.DATE
TemporalType.TIME

değerlerinden uygun olan
birisi kullanılabilir

Date ve Time Tipleri

- Hibernate **java.util.Date** tipinde bir değişkene ait değeri DB'den yüklerken **@Temporal** tanımına göre **java.sql.Timestamp**, **java.sql.Time**, veya **java.sql.Date** tipinde bir değer olarak yükler

```
@Entity
public class Pet {

    @Id
    @GeneratedValue
    private Long id;

    @Temporal(TemporalType.DATE)
    private java.util.Date birthDate;

    ...
}
```

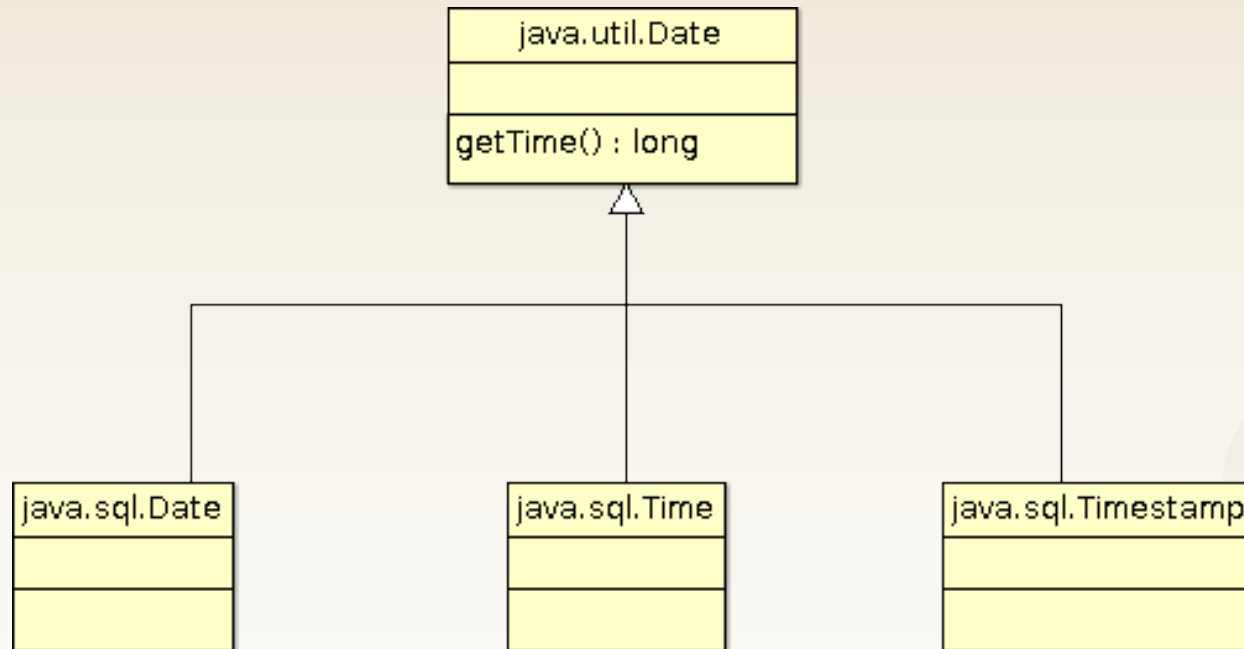
Entity içerisinde **java.util.Date** tipinde tanımlı bir property değeri runtime'da Entity DB'den yüklendiği vakit değer **java.sql.Date** tipinde yüklenecektir

```
Pet pet = session.get(Pet.class, 1L);
java.sql.Date birthDate = (Date) pet.getBirthDate();
```

Date ve Time Tipleri

Java Tipi	Eşleme Tipi	SQL Tipi
java.util.Date / java.sql.Timestamp	timestamp	TIMESTAMP
java.util.Calendar	calendar	TIMESTAMP
java.util.Date / java.sql.Date	date	DATE
java.util.Calendar	calendar_date	DATE
java.util.Date / java.sql.Time	time	TIME

Date ve Time Tipleri



- `java.sql.Date`, `java.sql.Time` ve `java.sql.Timestamp` sınıfları `java.util.Date` sınıfından türerler

Date ve Time Tipleri

- java.util.Date ile java.sql.Timestamp equals() metotları **simetrik olmadığından** bu durum nesnelerin **equals metot implemantasyonlarında soruna yol açabilir**



```
SimpleDateFormat df = new SimpleDateFormat("dd/MM/yyyy");
```

```
Date date = df.parse("01/01/1970");
```

```
long val = date.getTime();
```

```
Timestamp timestamp = new Timestamp(val);
```

Bu iki nesne değer olarak birbirinin aynısıdır!

```
System.out.println(date.equals(timestamp));
```

```
System.out.println(timestamp.equals(date));
```

```
true
```

```
false
```

java.sql.Timestamp Sınıfının equals Metodu

```
public class Timestamp {
```

```
...
```

```
public boolean equals(java.lang.Object ts) {  
    if (ts instanceof Timestamp) {  
        return this.equals((Timestamp)ts);  
    } else {  
        return false;  
    }  
}
```

```
public boolean equals(Timestamp ts) {  
    if (super.equals(ts)) {  
        if (nanos == ts.nanos) {  
            return true;  
        } else {  
            return false;  
        }  
    } else {  
        return false;  
    }  
}
```



Timestamp içerisinde nanosaniye düzeyinde bilgi ayrıca tutulduğu için equals metoduna parametre gelen nesne Timestamp ise bu tipe downcast edilerek iş Timestamp sınıfındaki diğer bir equals metoduna delege edilmektedir. eğer input argüman Timestamp değil ise hemen false dönlümüştür!

Date Tipinde Değer İçeren Equals Metotlarını Yazmak

```
@Entity
public class Pet {
    private String name;
    private Date birthDate;

    @Override
    public boolean equals(Object obj) {
        if (this == obj)
            return true;
        if (obj == null)
            return false;
        if (getClass() != obj.getClass())
            return false;
        Pet other = (Pet) obj;
        return name.equals(other.name)
            && birthDate.equals(other.birthDate);
    }
}
```

Yanlış!

Date Tipinde Değer İçeren Equals Metotlarını Yazmak

```
@Entity
public class Pet {
    private String name;
    private Date birthDate;

    @Override
    public boolean equals(Object obj) {
        if (this == obj)
            return true;
        if (obj == null)
            return false;
        if (getClass() != obj.getClass())
            return false;
        Pet other = (Pet) obj;
        return name.equals(other.name)
            && (birthDate.getTime() == other.birthDate.getTime());
    }
}
```

Doğru!

Hibernate ve LocalDate

- Java 8 ile birlikte gelen **Date ve Time API**'deki tipler **Hibernate 5** tarafından desteklenmektedir
- Duration, Instant, LocalDate, LocalDateTime, LocalTime, OffsetTime gibi Java tipleri doğrudan entity attribute'larında kullanılabilir
- **@Temporal** anotasyonunu **kullanmaya gerek yoktur**

```
@Entity
public class Movie {
    @Id
    private Long id;

    private java.time.Duration duration;

    ...
}
```

Java 8 Date Time API ORM Tipleri

Java Tipi	Eşleme Tipi	SQL Tipi
java.time.Duration	long	BIGINT
java.time.LocalDateTime	timestamp	TIMESTAMP
java.time.LocalDate	date	DATE
java.time.LocalTime	time	TIME
java.time.OffsetDateTime	timestamp	TIMESTAMP
java.time.OffsetTime	time	TIME
java.time.ZonedDateTime	timestamp	TIMESTAMP

JPA ve LocalDate

- **LocalDate** Java 8 ile gelen bir kabiliyettir
- **JPA 2.1** Java 8 öncesinde çıktığı için **LocalDate** desteği yoktur
- **LocalDate** tipinde property'leri kullanabilmek için **AttributeConverter**'dan yararlanılabilir

JPA ve LocalDate

```
import java.sql.Date;
import java.time.LocalDate;

import javax.persistence.AttributeConverter;
import javax.persistence.Converter;

@Converter(autoApply = true)
public class LocalDateAttributeConverter
    implements AttributeConverter<LocalDate, Date> {

    @Override
    public Date convertToDatabaseColumn(LocalDate localDate) {
        return (localDate == null ? null : Date.valueOf(localDate));
    }

    @Override
    public LocalDate convertToEntityAttribute(Date sqlDate) {
        return (sqlDate == null ? null : sqlDate.toLocalDate());
    }
}
```

İletişim

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

