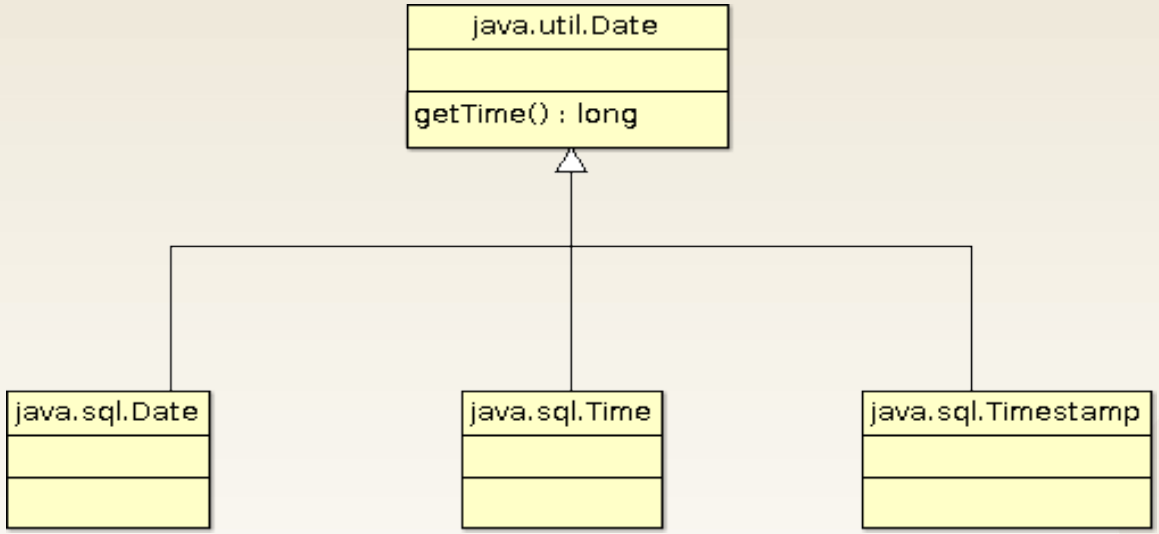
JDBC ve Temporal Tipler

harezmi bilişim çözümleri

JDBC API ve java.sql.Date/Time/Timestamp



- **JDBC API** ile birlikte gelen Date, Time ve Timestamp sınıfları da mevcuttur
- java.sql.Date, Time ve Timestamp sınıfları java.util.Date sınıfından türerler
- Timestamp sınıfı **nanosaniye** düzeyinde zaman bilgisini ele almayı sağlar

JDBC API ve java.sql.Date/Time/Timestamp

- JDBC API'deki **ResultSet** sınıfının **getDate**, **getTime** ve **getTimestamp** metotları bu sınıflardan nesneler döner

Date ve Timestamp Arasındaki Uyumsuzluk

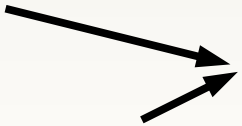
- java.util.Date ile java.sql.Timestamp equals() metotları **simetrik olmadığından** bu durum nesnelerin **equals metot implemantasyonlarında soruna** yol açabilir

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

Date date = df.parse("01/01/1970");
long val = date.getTime();
Timestamp timestamp = new Timestamp(val);

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

true
false
```



Bu iki nesne değer olarak birbirinin aynısıdır!

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 dönülmüştür!

Date Tipinde Değer İçeren Equals Metotlarını Yazmak

```
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);
    }
}
```

Date Tipinde Değer İçeren Equals Metotlarını Yazmak

```
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());
    }
}
```

JDBC ve Java 8 Date/Time API

Java 8 Date/Time API'deki tiplerin SQL tip karşılıkları

Java 8	ANSI SQL
LocalDate	DATE
LocalTime	TIME
LocalDateTime	TIMESTAMP
OffsetTime	TIME with TIMEZONE
OffsetDateTime	TIMESTAMP with TIMEZONE

JDBC ve Java 8 Date/Time API

- Eğer JDBC sürücü Java 8 Date/Time API'sini **desteklemiyor ise** veri işlemleri java.sql.Date, java.sql.Time ve java.sql.Timestamp tiplerini kullanarak yapılmalıdır

```
Statement stmt = connection.createStatement(
                 ResultSet.TYPE_FORWARD_ONLY, ResultSet.CONCUR_UPDATABLE);

ResultSet rs = stmt.executeQuery("select * from T_PET");

while (rs.next()) {
    java.sql.Date date = rs.getDate("BIRTH_DATE");
    LocalDate localDate = date.toLocalDate();
    System.out.println(localDate);
    rs.updateObject("BIRTH_DATE", java.sql.Date.valueOf(LocalDate.now()));
    rs.updateRow();
}
```

JDBC ve Java 8 Date/Time API

- Eğer JDBC sürücüsü Java 8 Date/Time API ile **uyumlu ise** doğrudan yeni tiplerle çalışılabilir

```
Statement stmt = connection.createStatement(
           ResultSet.TYPE_FORWARD_ONLY, ResultSet.CONCUR_UPDATABLE);

ResultSet rs = stmt.executeQuery("select * from T_PET");

while (rs.next()) {
    LocalDate localDate = rs.getObject("BIRTH_DATE", LocalDate.class);
    System.out.println(localDate);
    rs.updateObject("BIRTH_DATE", LocalDate.now());
    rs.updateRow();
}
```

İletişim

www.harezmi.com.tr

www.java-egitimleri.com

info@harezmi.com.tr

info@java-egitimleri.com

@HarezmiBilisim

@JavaEgitimleri