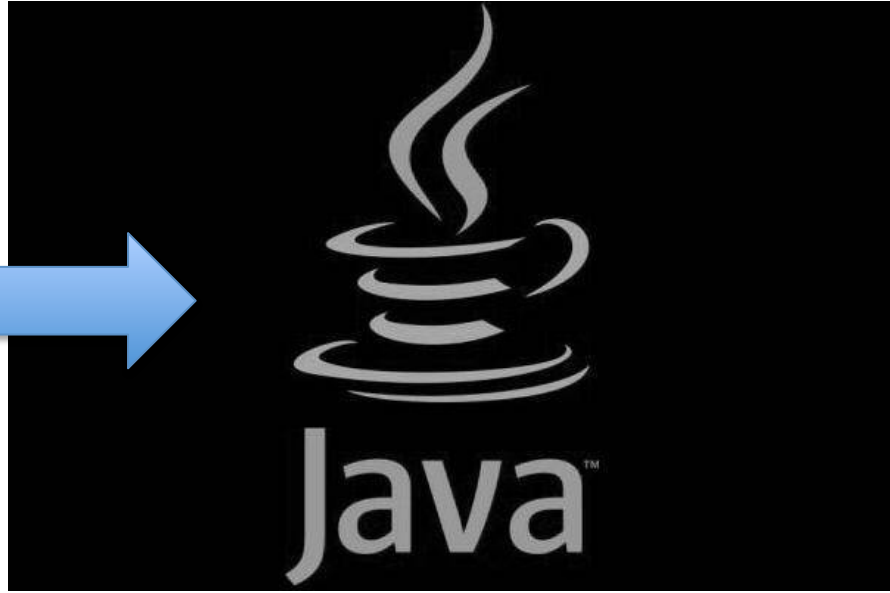
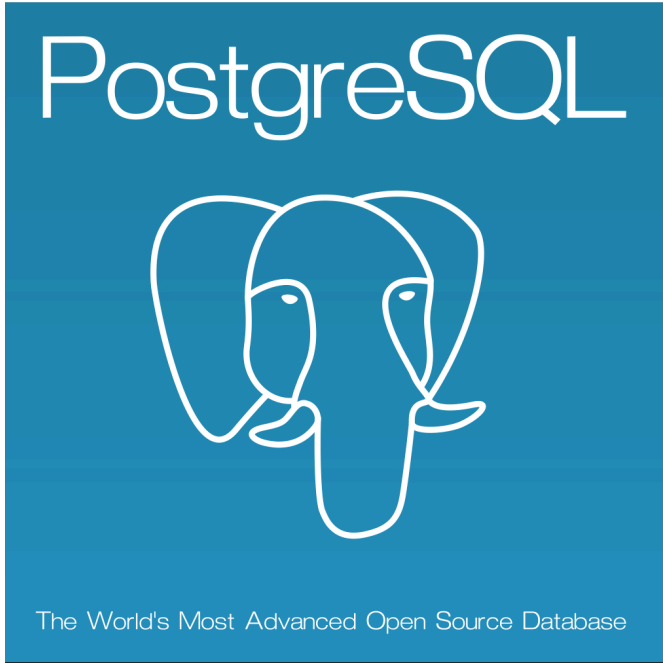


# JDBC

Java Data Base Connectivity



Package java.sql

# Package java.sql

Array

Blob

CallableStatement

Clob

Connection

DatabaseMetaData

Driver

NClob

ParameterMetaData

PreparedStatement

Ref

ResultSet

ResultSetMetaData

RowId

Savepoint

SQLData

SQLInput

SQLOutput

SQLXML

Statement

Struct

Wrapper

# Package java.sql

Array

Blob

CallableStatement

Clob

**Connection**

DatabaseMetaData

Driver

NClob

ParameterMetaData

PreparedStatement

Ref

ResultSet

ResultSetMetaData

RowId

Savepoint

SQLData

SQLInput

SQLOutput

SQLXML

**Statement**

Struct

Wrapper

# Package java.sql

7 klasser i java.sql. Vi trenger én av dem:

DriverManager

The basic service for managing a set of JDBC drivers

Her finner vi metoder som gir oss en forbindelse. Dette skjer med metoder som returnerer (en peker til) et objekt av typen Connection.

# DriverManager

Her finner vi metoder som gir oss en forbindelse. Dette skjer med metoder som returnerer (en peker til) et objekt av typen Connection.

```
static Connection getConnection(String url, Properties info)
```

# DriverManager

Her finner vi metoder som gir oss en forbindelse. Dette skjer med metoder som returnerer (en peker til) et objekt av typen Connection.

```
static Connection getConnection(String url, Properties info)
```

```
DriverManager.getConnection(url, p);
```



```
DriverManager.getConnection(url, p);
```

```
DriverManager.getConnection(url, p);
```

```
String url = "jdbc:postgresql://dbpg-ifi-kurs.uio.no/fdb";
```

```
DriverManager.getConnection(url, p);
```

```
String url = "jdbc:postgresql://dbpg-ifi-kurs.uio.no/fdb";
```

```
Properties p = new Properties();
```

```
java.util
```

## **Class Properties**

```
java.lang.Object
```

```
java.util.Dictionary<K,V>
```

```
java.util.Hashtable<Object,Object>
```

```
java.util.Properties
```

### **All Implemented Interfaces:**

```
Serializable, Cloneable, Map<Object,Object>
```

### **Direct Known Subclasses:**

```
Provider
```

---

```
public class Properties  
extends Hashtable<Object,Object>
```

```
DriverManager.getConnection(url, p);
```

```
String url = "jdbc:postgresql://dbpg-ifi-kurs.uio.no/fdb";
```

```
Properties p = new Properties();
```

```
p.setProperty("ssl", "true");
```

```
DriverManager.getConnection(url, p);
```

```
String url = "jdbc:postgresql://dbpg-ifi-kurs.uio.no/fdb";
```

```
Properties p = new Properties();
```

```
p.setProperty("ssl", "true");
```

```
p.setProperty("sslfactory",  
              "org.postgresql.ssl.NonValidatingFactory");
```

```
DriverManager.getConnection(url, p);
```

```
String url = "jdbc:postgresql://dbpg-ifi-kurs.uio.no/fdb";
```

```
Properties p = new Properties();
```

```
p.setProperty("ssl", "true");
```

```
p.setProperty("sslfactory",  
              "org.postgresql.ssl.NonValidatingFactory");
```

```
p.put("user", "michael");
```

```
DriverManager.getConnection(url, p);
```

```
String url = "jdbc:postgresql://dbpg-ifi-kurs.uio.no/fdb";
```

```
Properties p = new Properties();
```

```
p.setProperty("ssl", "true");
```

```
p.setProperty("sslfactory",  
              "org.postgresql.ssl.NonValidatingFactory");
```

```
p.put("user", "michael");
```

```
p.put("password", password);
```

```
DriverManager.getConnection(url, p);
```

```
String url = "jdbc:postgresql://dbpg-ifi-kurs.uio.no/fdb";
```

```
Properties p = new Properties();
```

```
p.setProperty("ssl", "true");
```

```
p.setProperty("sslfactory",  
              "org.postgresql.ssl.NonValidatingFactory");
```

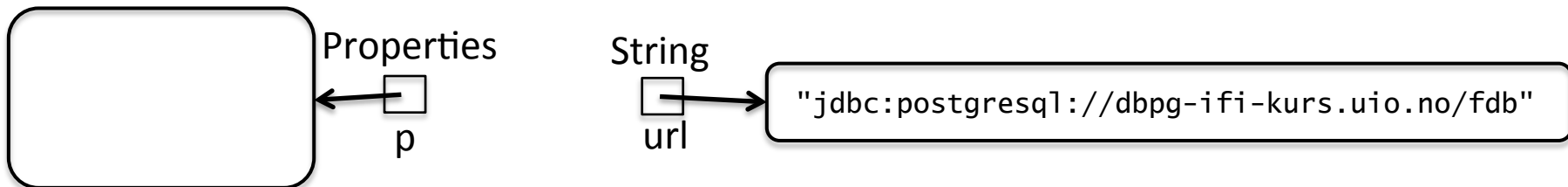
```
p.put("user", "michael");
```

```
p.put("password", password);
```



```
DriverManager.getConnection(url, p);
```

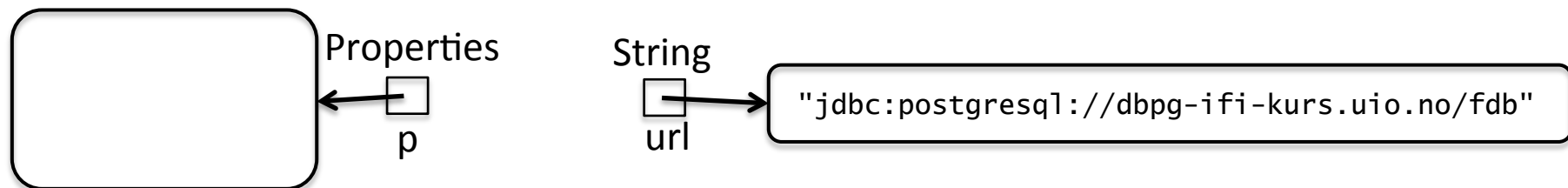
```
String url = "jdbc:postgresql://dbpg-ifi-kurs.uio.no/fdb";  
  
Properties p = new Properties();  
  
p.setProperty("ssl", "true");  
  
p.setProperty("sslfactory",  
              "org.postgresql.ssl.NonValidatingFactory");  
  
p.put("user", "michael");  
  
p.put("password", password);
```



DriverManager.getConnection(url, p);

*Klassedatastruktur for klassen DriverManager*

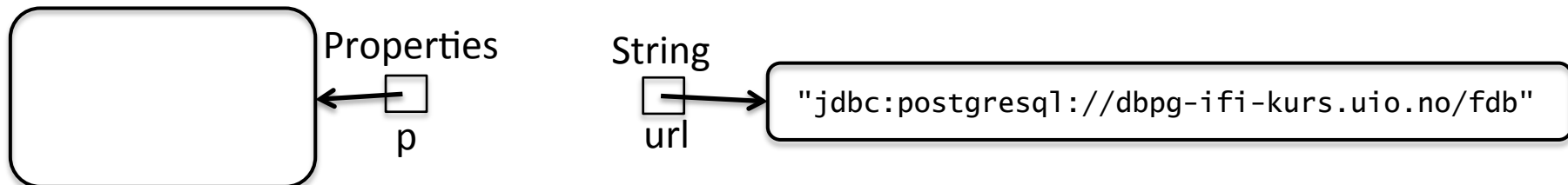
static Connection getConnection(String url, Properties info)



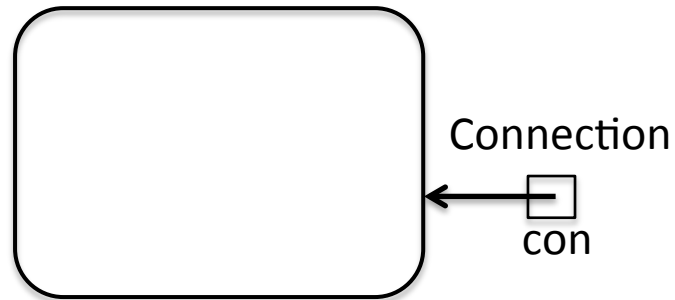
```
Connection con = DriverManager.getConnection(url, p);
```

*Klassedatastruktur for klassen DriverManager*

```
static Connection getConnection(String url, Properties info)
```



```
Connection con = DriverManager.getConnection(url, p);
```



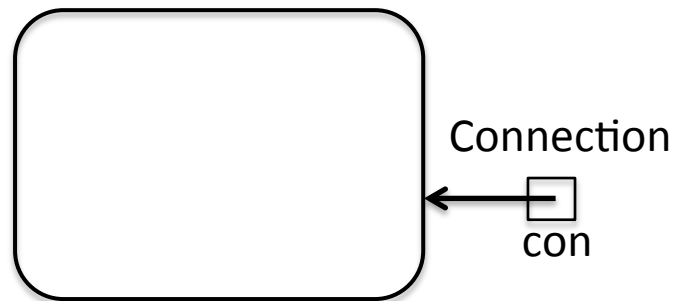
# DriverManager

Her finner vi metoder som gir oss en forbindelse. Dette skjer med metoder som returnerer (en peker til) et objekt av typen Connection.

```
static Connection getConnection(String url, Properties info)
```

# Connection

public interface Connection extends Wrapper, AutoCloseable

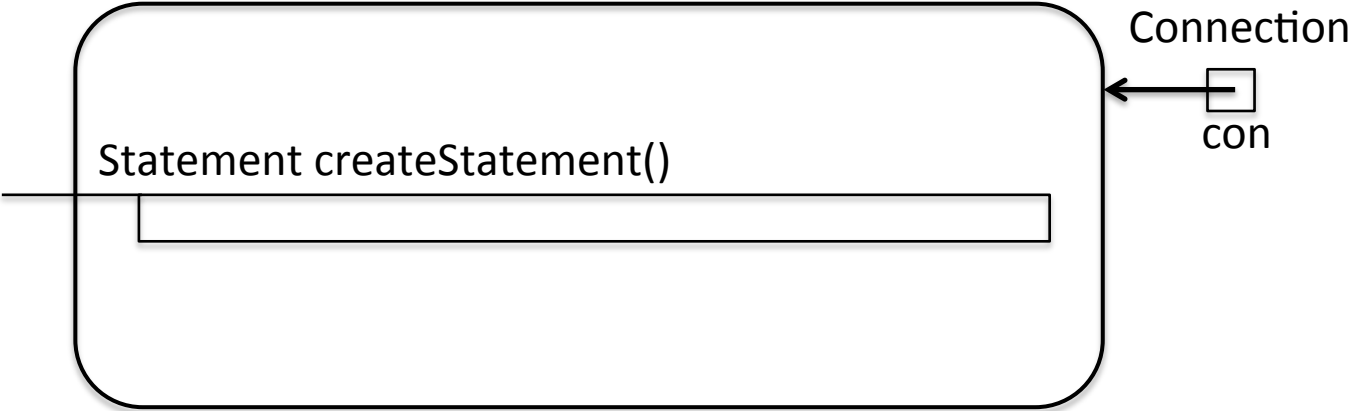


# DriverManager

Her finner vi metoder som gir oss en forbindelse. Dette skjer med metoder som returnerer (en peker til) et objekt av typen Connection.

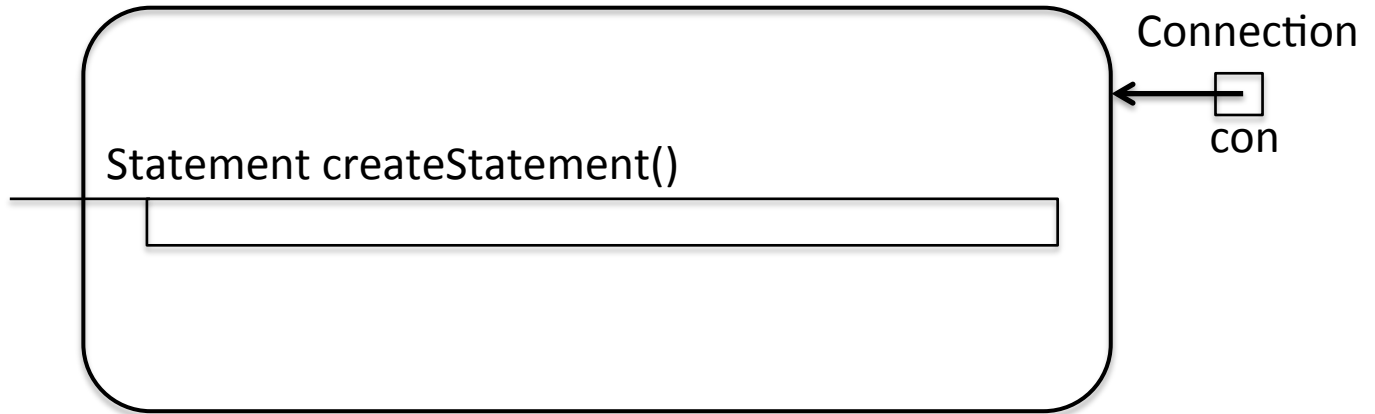
```
static Connection getConnection(String url, Properties info)
```

# Et Connection-objekt



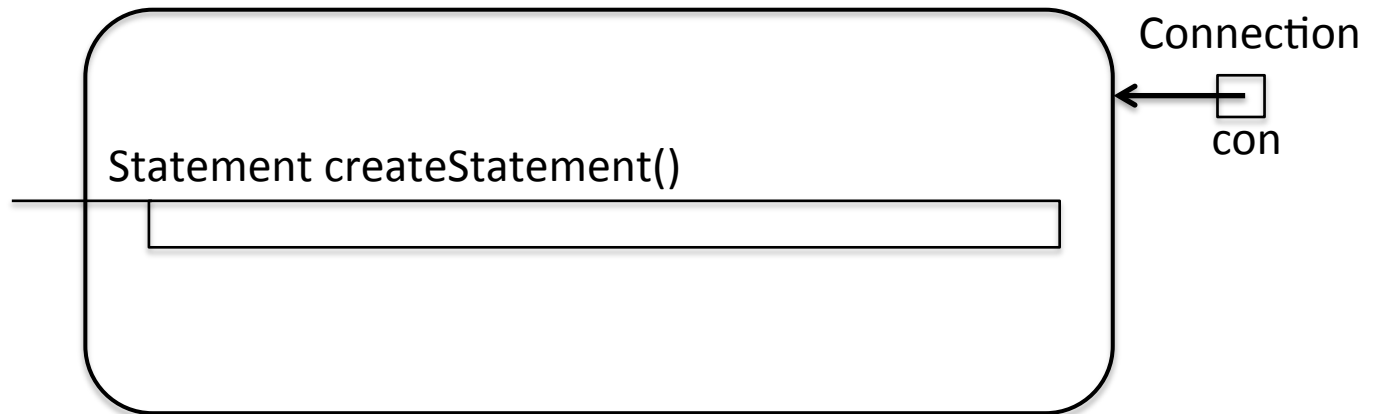


# Et Connection-objekt



`/* Creates a Statement object for sending  
SQL statements to the database. */`

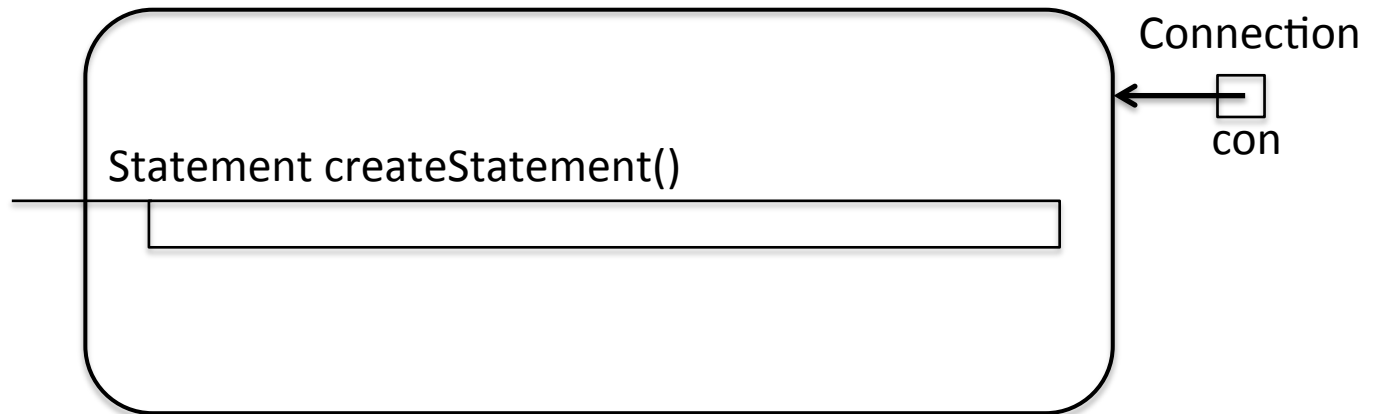
# Et Connection-objekt



```
/* Creates a Statement object for sending  
SQL statements to the database. */
```

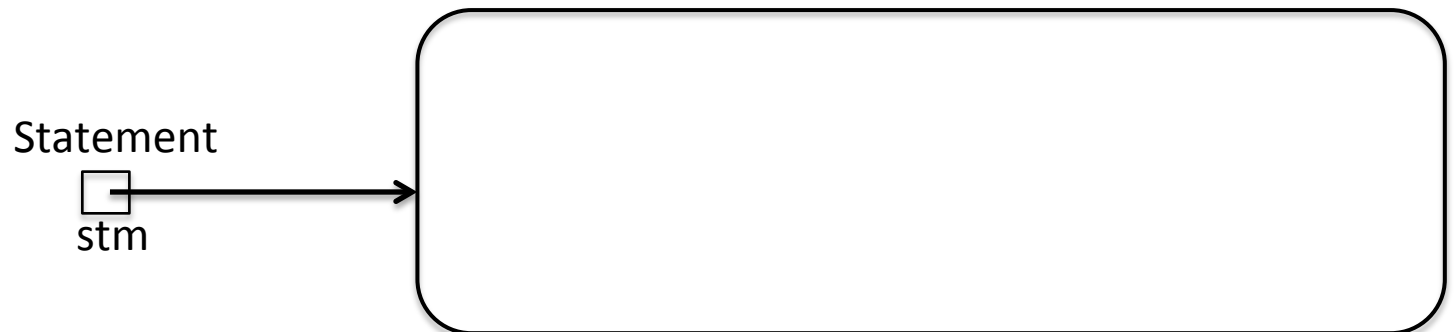
```
Statement stm = con.createStatement();
```

# Et Connection-objekt



`/* Creates a Statement object for sending  
SQL statements to the database. */`

```
Statement stm = con.createStatement();
```



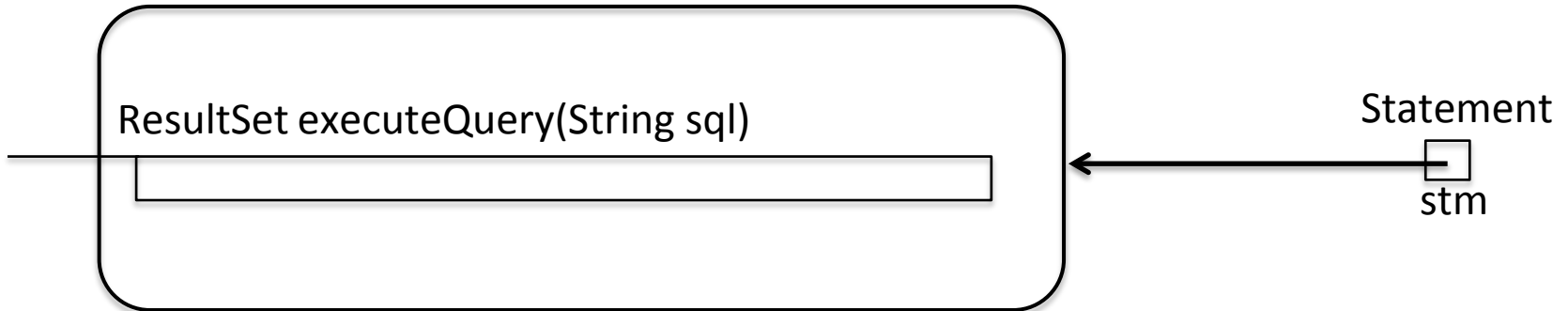
# Et Statement-objekt



# Et Statement-objekt

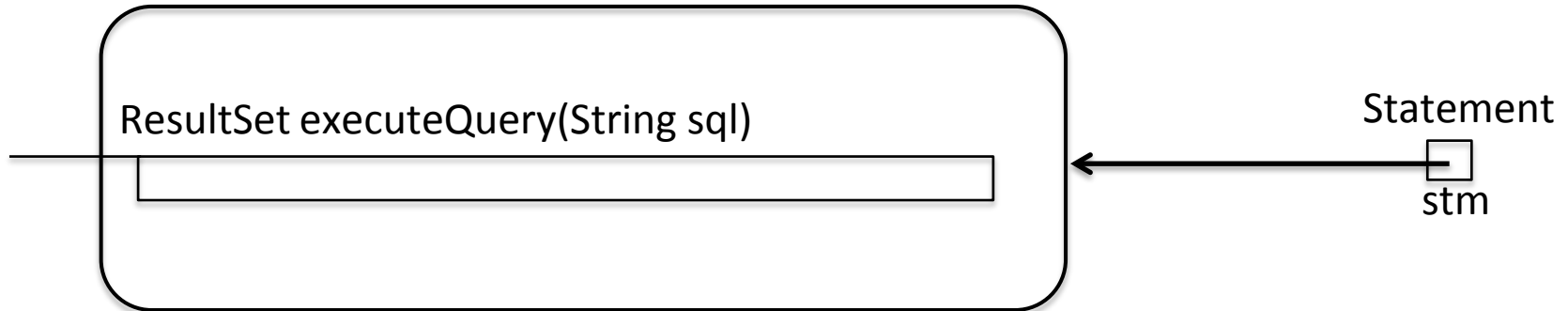


# Et Statement-objekt



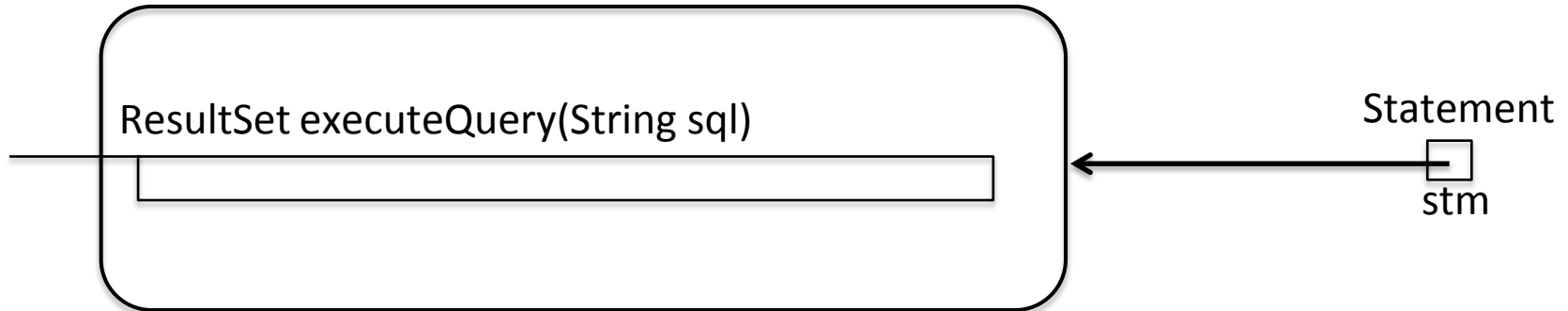
Executes the given SQL statement,  
which returns a single ResultSet object.

# Et Statement-objekt



String spørring = " select F.filmid as fid, F.title as tittel, F.prodyear as paar  
from film F natural join Filmcountry C where C.country='Norway' " ;

# Et Statement-objekt

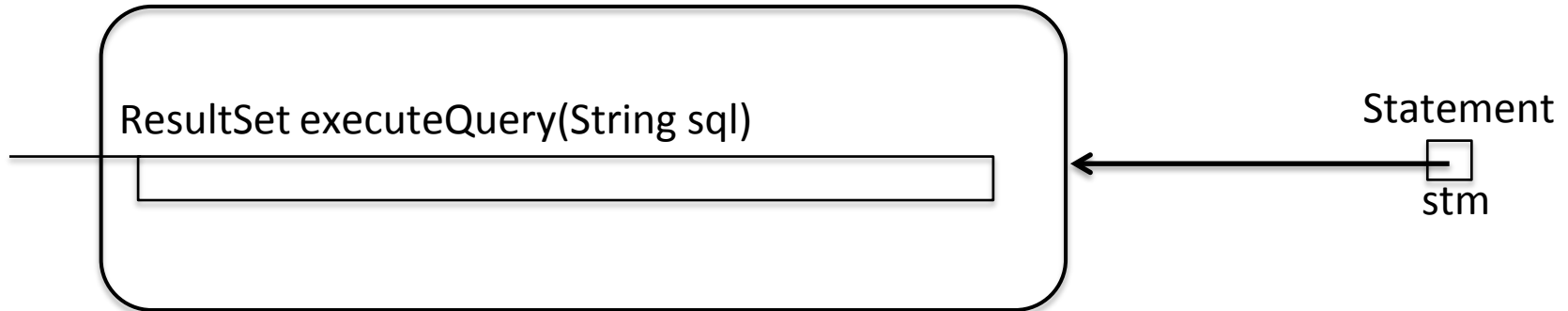


```
String spørring = " select F.filmid as fid, F.title as tittel, F.prodyear as paar  
from film F natural join Filmcountry C where C.country='Norway' " ;
```

```
String spørring = "select F.filmid as fid, "  
+"      F.title as tittel, "  
+"      F.prodyear as paar "  
+"from film F natural join Filmcountry C "  
+"where C.country='Norway' " ;
```

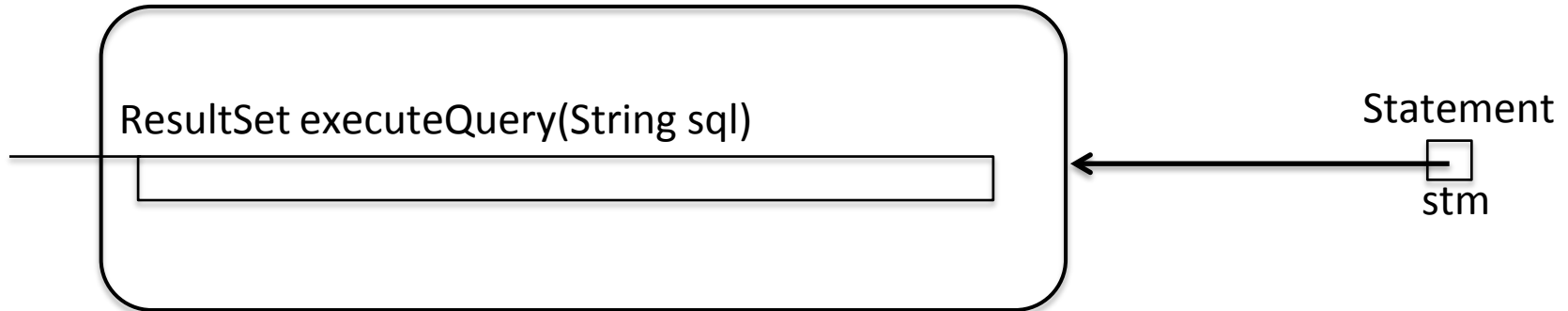


# Et Statement-objekt



```
String spørring = "select F.filmid as fid, "  
                  +"          F.title as tittel, "  
                  +"          F.prodyear as paar "  
                  +"from film F natural join Filmcountry C "  
                  +"where C.country='Norway' " ;
```

# Et Statement-objekt



```
String spørring = "select F.filmid as fid, "  
                  +"          F.title as tittel, "  
                  +"          F.prodyear as paar "  
                  +"from film F natural join Filmcountry C "  
                  +"where C.country='Norway' " ;
```

```
ResultSet norskeFilmer = stm.executeQuery (spørring);
```

# ResultSet

fid	tittel	paar
230	Varis	2004
340	Anolit	2002
356	Kvinnen i mitt liv	2003
632	Syx	1988
664	Portrettet	1954
774	22	2000
792	Andre omgang	2007
998	Digre daier	1997
1014	Stopp	2001
1030	Tid for frokost	2004
...		

ResultSet  
norskeFilmer

```
ResultSet norskeFilmer = stm.executeQuery (spørring);
```

# ResultSet

fid	tittel	paar
230	Varis	2004
340	Anolit	2002
356	Kvinnen i mitt liv	2003
632	Syx	1988
664	Portrettet	1954
774	22	2000
792	Andre omgang	2007
998	Digre daier	1997
1014	Stopp	2001
1030	Tid for frokost	2004
...		

ResultSet

norskeFilmer

*A table of data representing a database result set, which is usually generated by executing a statement that queries the database.*

```
ResultSet norskeFilmer = stm.executeQuery (spørring);
```

# ResultSet

fid	tittel	paar
230	Varis	2004
340	Anolit	2002
356	Kvinnen i mitt liv	2003
632	Syx	1988
664	Portrettet	1954
774	22	2000
792	Andre omgang	2007
998	Digre daier	1997
1014	Stopp	2001
1030	Tid for frokost	2004
...		

ResultSet

←  norskeFilmer

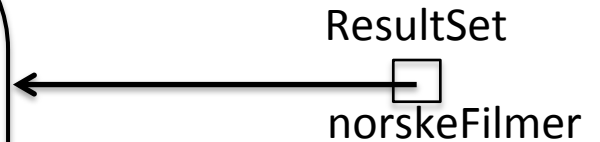
*A ResultSet object maintains a cursor pointing to its current row of data. Initially the cursor is positioned before the first row. The next method moves the cursor to the next row, and because it returns false when there are no more rows in the ResultSet object, it can be used in a while loop to iterate through the result set..*

```
ResultSet norskeFilmer = stm.executeQuery (spørring);
```

# ResultSet

fid	tittel	paar
230	Varis	2004
340	Anolit	2002
356	Kvinnen i mitt liv	2003
632	Syx	1988
664	Portrettet	1954
774	22	2000
792	Andre omgang	2007
998	Digre daier	1997
1014	Stopp	2001
1030	Tid for frokost	2004
...		

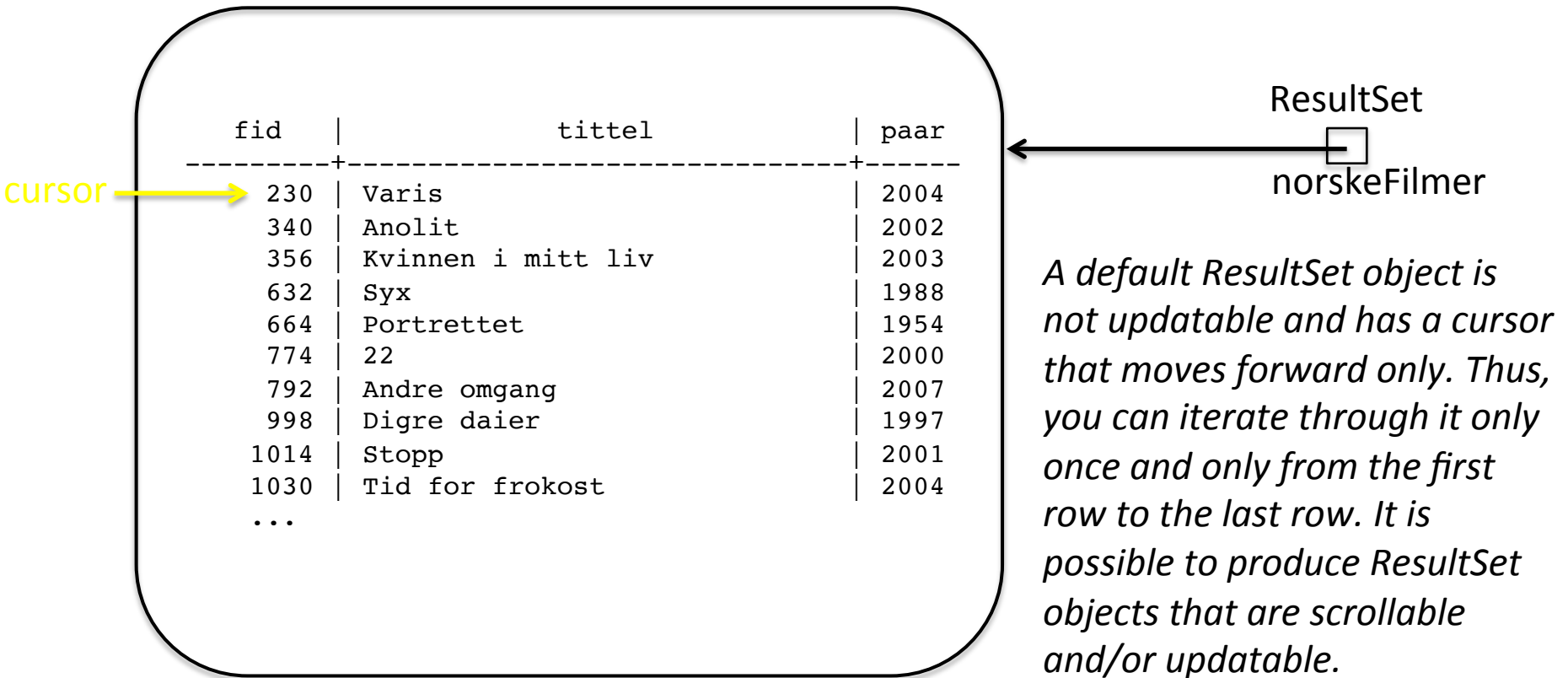
A yellow arrow labeled "cursor" points to the first row of the table.



*A ResultSet object maintains a **cursor** pointing to its current row of data. Initially the cursor is positioned before the first row. The next method moves the cursor to the next row, and because it returns false when there are no more rows in the ResultSet object, it can be used in a while loop to iterate through the result set..*

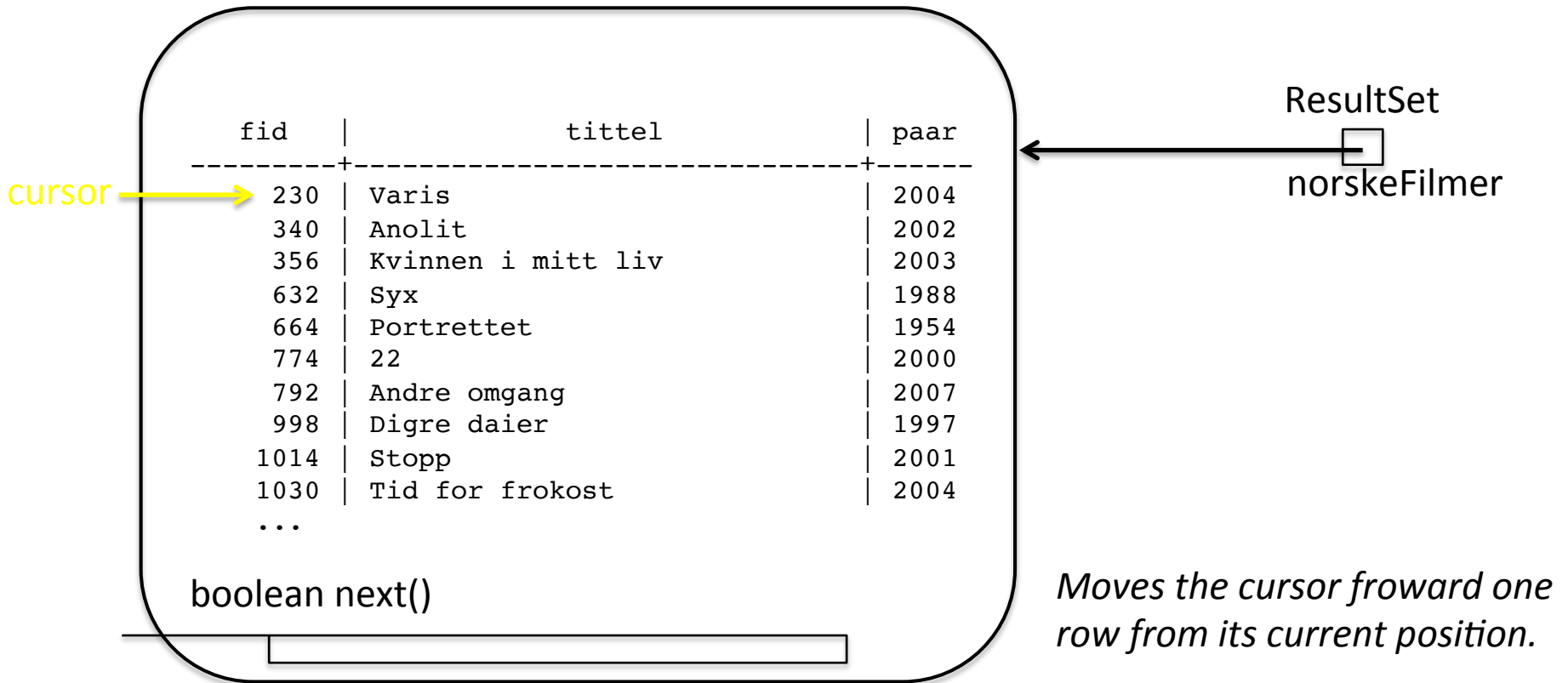
```
ResultSet norskeFilmer = stm.executeQuery (spørring);
```

# ResultSet



```
ResultSet norskeFilmer = stm.executeQuery (spørring);
```

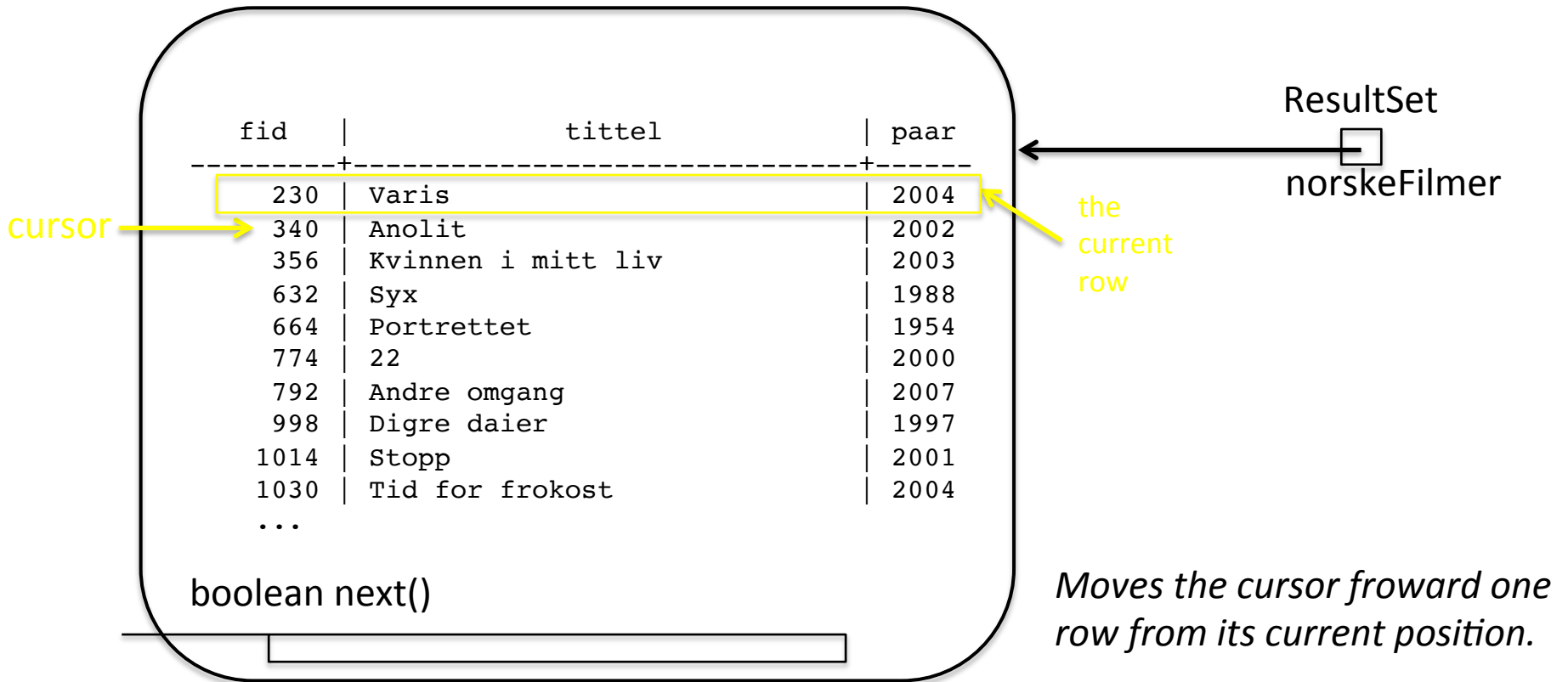
# ResultSet



```
ResultSet norskeFilmer = stm.executeQuery (spørring);
```

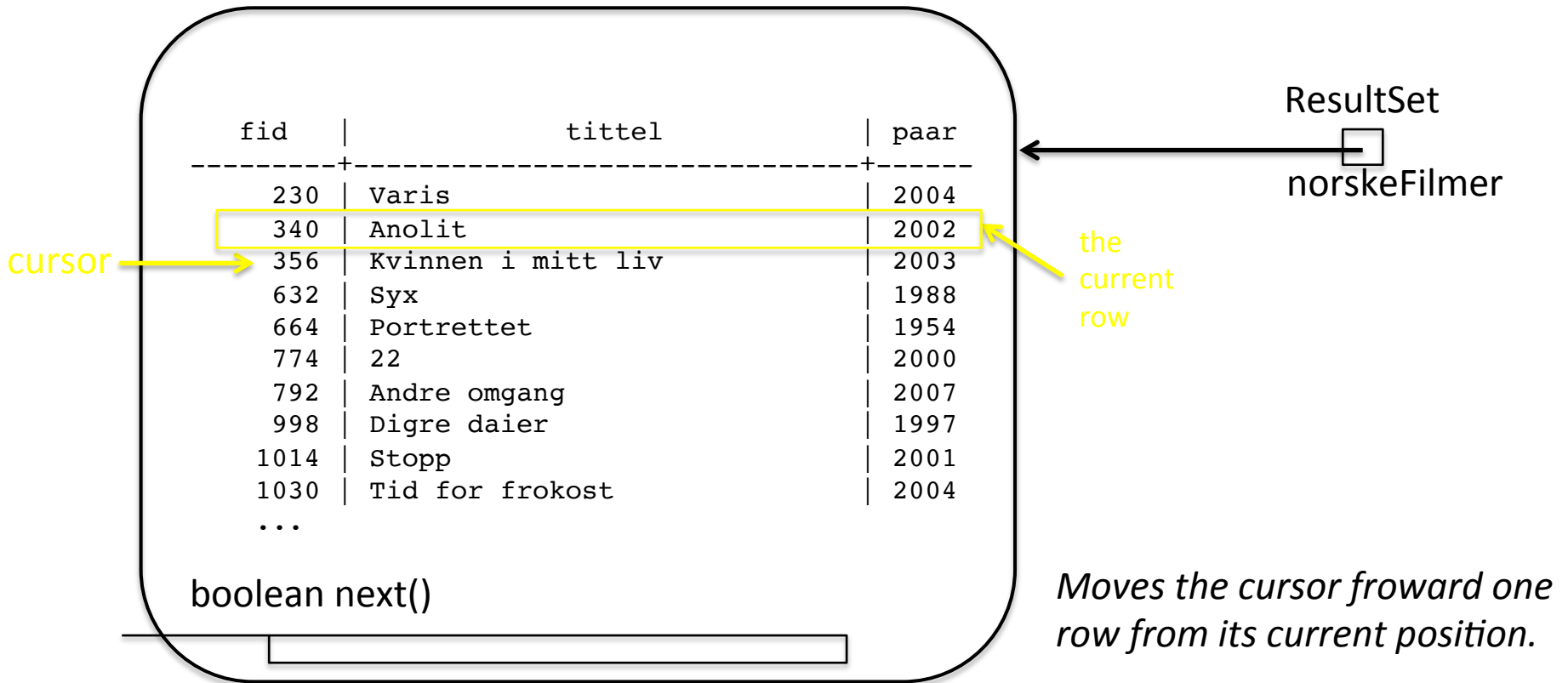


# ResultSet



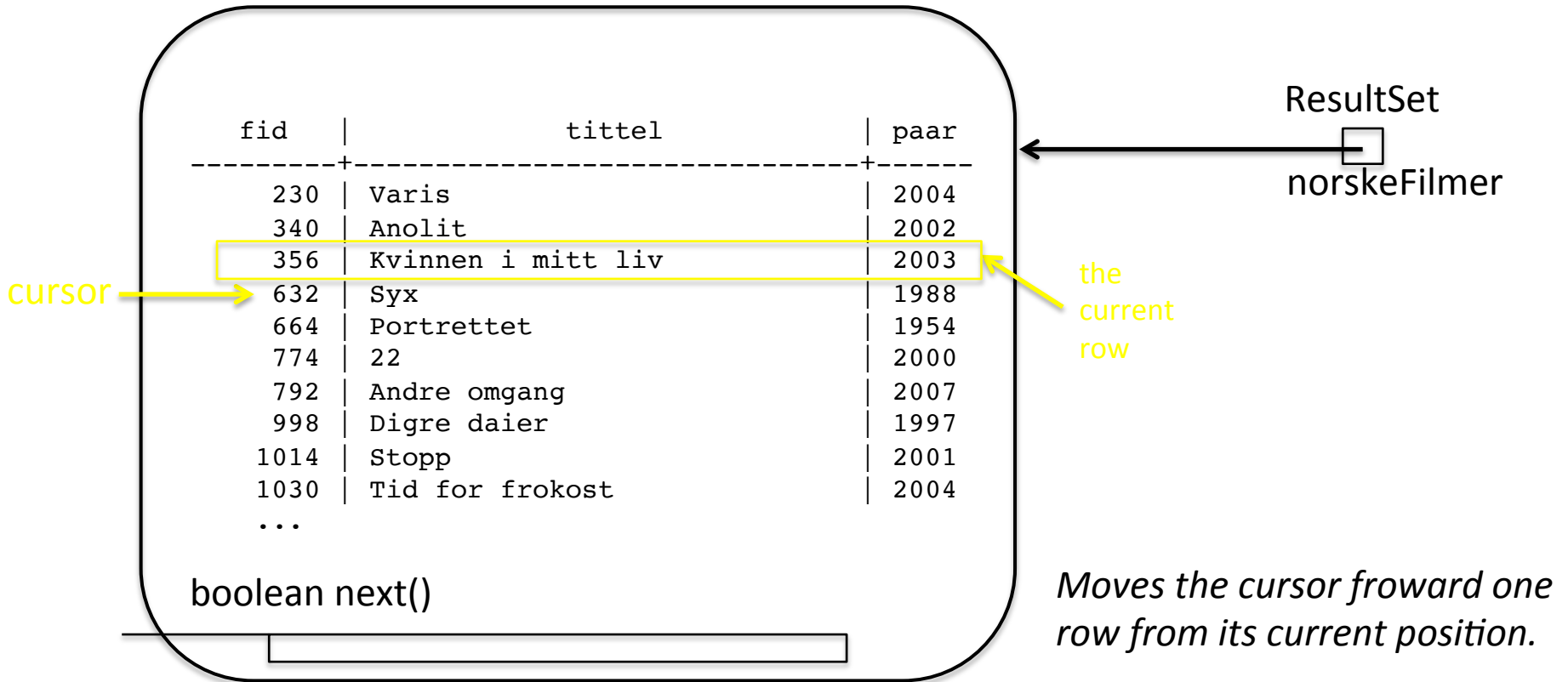
```
ResultSet norskeFilmer = stm.executeQuery (spørring);
```

# ResultSet



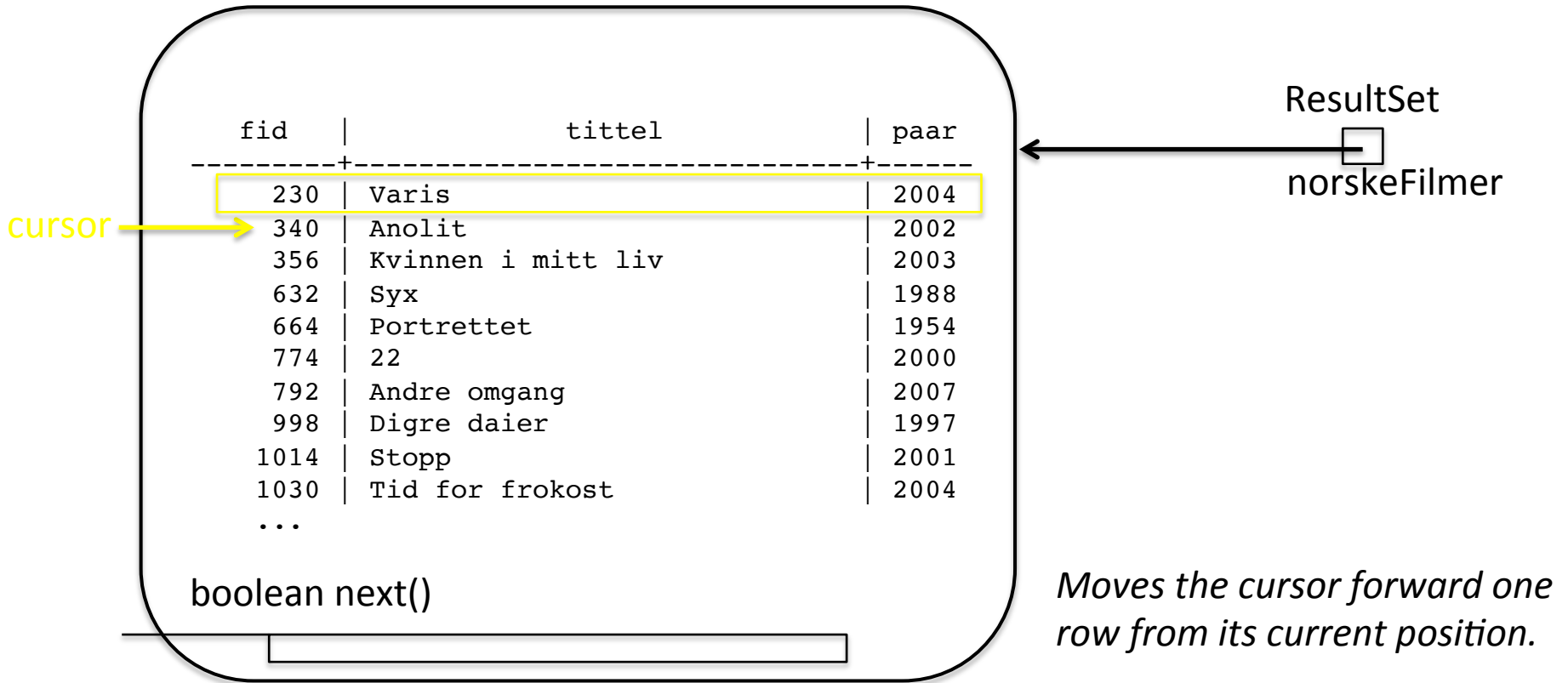
```
ResultSet norskeFilmer = stm.executeQuery (spørring);
```

# ResultSet



```
ResultSet norskeFilmer = stm.executeQuery (spørring);
```

# ResultSet



```
ResultSet norskeFilmer = stm.executeQuery (spørring);
```

# ResultSet

fid	tittel	paar
230	Varis	2004
340	Anolit	2002
356	Kvinnen i mitt liv	2003
632	Syx	1988
664	Portrettet	1954
774	22	2000
792	Andre omgang	2007
998	Digre daier	1997
1014	Stopp	2001
1030	Tid for frokost	2004
...		

boolean next()

String getString(String columnLabel)

int getInt(int columnIndex)

ResultSet

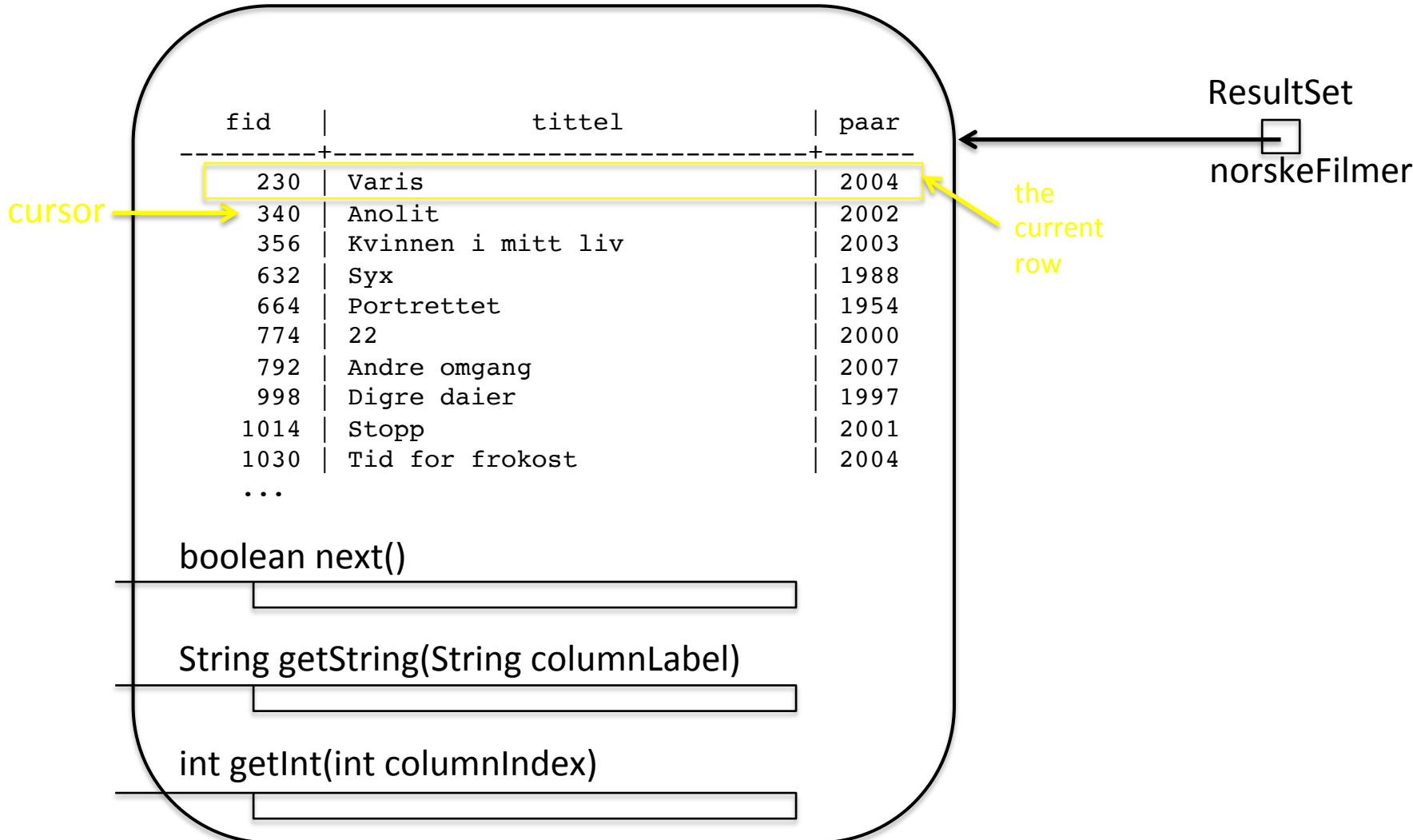
norskeFilmer

the  
current  
row

*Moves the cursor forward one row from its current position.*

*Retrieves the value of the designated column in the current row of this ResultSet object as a String.*

*Retrieves the value of the designated column in the current row of this ResultSet object as an int.*



ResultSet  
 norskeFilmer

cursor

the current row

```
while ( norskeFilmer.next() ) {
    fm = new Film(norskeFilmer.getInt("fid"));
    fm.tittel = norskeFilmer.getString("tittel");
    fm.produksjonsaar = norskeFilmer.getInt("paar");
    filmer.add(fm);
}
```

```

Properties p = new Properties();
p.setProperty("ssl", "true");
p.setProperty("sslfactory", "org.postgresql.ssl.NonValidatingFactory");
p.put("user", "michael");
p.put("password", passord); // "ifikurs_ro");
String url = "jdbc:postgresql://dbpg-ifi-kurs.uio.no/fdb";

Connection con = DriverManager.getConnection(url, p); }

Statement stm = con.createStatement();

String spørring = "select F.filmid as fid, "
                  +"          F.title as tittel, "
                  +"          F.prodyear as paar "
                  +"from film F natural join Filmcountry C "
                  +"where C.country='Norway'" ;

ResultSet norskeFilmer = stm.executeQuery (spørring);

while (norskeFilmer.next() ) {
    fm = new Film(norskeFilmer.getInt("fid"));
    fm.tittel = norskeFilmer.getString("tittel");
    fm.produksjonsår = norskeFilmer.getInt("paar");
    filmer.add(fm);
}

```