



```

7 import java.sql.Connection;
8 import java.sql.DriverManager;
9 import java.sql.ResultSet;
10 import java.sql.ResultSetMetaData;
11 import java.sql.SQLException;
12 import java.sql.Statement;
    
```

```

14 public class JDBCExample {
15     public static void outputResultSet(ResultSet rset) throws SQLException{
16         ResultSetMetaData rsmd = rset.getMetaData();
17         final int numberOfColumns = rsmd.getColumnCount();
18         while (rset.next()) {
19             for (int i = 1; i <= numberOfColumns; i++) {
20                 System.out.print(rset.getObject(i) + "\t");
21             }
22             System.out.println();
23         }
24     }
25 }
    
```

```

27 public static void main(String[] args) {
28     Connection conn = null;
29
30     try {
31         System.out.println(Class.forName("org.postgresql.Driver"));
32         conn = DriverManager.getConnection("jdbc:postgresql://localhost/Fotoagentur?user=postgres&password=.....");
33     } catch (Exception e) {
34         System.out.println("Fehler: " + e);
35         System.exit(-1);
36     }
37     if (conn != null) {
38         try {
39             Statement sql_stmt = conn.createStatement();
40             ResultSet rset = sql_stmt.executeQuery("select * from FotografenKomplett2");
41             outputResultSet(rset);
42             rset.close();
43             conn.close();
44         } catch (SQLException se) {
45             System.out.println("Fehler: " + se);
46         }
47     }
48 }
49 }
    
```



Postgres

3	Mueller Peter	1963-10-09	50000	2
6	Wurst Hans	1974-02-01	15000	1
7	Miese Peter	1983-05-06	50000	2
42	Froehlich Frida	1987-03-02	52000	2
77	Sorglos Lisa	1977-07-12	62000	3
34	Mueller Herta	1992-11-30	32000	1

BUILD SUCCESSFUL (total time: 0 seconds)

```

7 import java.sql.Connection;
8 import java.sql.DriverManager;
9 import java.sql.ResultSet;
10 import java.sql.ResultSetMetaData;
11 import java.sql.SQLException;
12 import java.sql.Statement;
13 import static jdbcexample.JDBCExample.outputResultSet;
14
15 public class JDBCParameterExample {
16
17     public static void outputResultSet(ResultSet rset) throws SQLException {...}
18
19     public static void main(String[] args) {
20         Connection conn = null;
21
22         try {
23             System.out.println(Class.forName("org.postgresql.Driver"));
24             conn = DriverManager.getConnection("jdbc:postgresql://localhost/Fotoagentur?user=postgres&password=.....");
25         } catch (Exception e) {
26             System.out.println("Fehler: " + e);
27             System.exit(-1);
28         }
29         if (conn != null) {
30             try {
31                 Statement sql_stmt = conn.createStatement();
32                 ResultSet rset = sql_stmt.executeQuery("select * from FotografenKomplett2 where id = " + args[0]);
33                 outputResultSet(rset);
34                 rset.close();
35                 conn.close();
36             } catch (SQLException se) {
37                 System.out.println("Fehler: " + se);
38             }
39         }
40     }
41 }
42
43 }
44
45 }
46
47 }
48
49 }
50
51 }

```

args[0] = 3



```

class org.postgresql.Driver
3      Mueller Peter    1963-10-09      50000      2
BUILD SUCCESSFUL (total time: 0 seconds)

```

```

7 import java.sql.Connection;
8 import java.sql.DriverManager;
9 import java.sql.ResultSet;
10 import java.sql.ResultSetMetaData;
11 import java.sql.SQLException;
12 import java.sql.Statement;
13
14 public class JDBCParameterExample {
15
16     public static void outputResultSet(ResultSet rset) throws SQLException {...}
17
18
19
20
21
22
23
24
25
26
27
28     public static void main(String[] args) {
29         Connection conn = null;
30
31         try {
32             System.out.println(Class.forName("org.postgresql.Driver"));
33             conn = DriverManager.getConnection("jdbc:postgresql://localhost/Fotoagentur?user=postgres&password=.....");
34         } catch (Exception e) {
35             System.out.println("Fehler: " + e);
36             System.exit(-1);
37         }
38         if (conn != null) {
39             try {
40                 Statement sql_stmt = conn.createStatement();
41                 final String queryString = "update Mitarbeiter2 set gehalt=gehalt*1.5 where personid = " + args[0];
42                 System.out.println(queryString);
43                 boolean success = sql_stmt.execute(queryString);
44                 conn.close();
45             } catch (SQLException se) {
46                 System.out.println("Fehler: " + se);
47             }
48         }
49     }
50 }

```

`args[0] = „3“`

personid	gehalt	erfahrung
integer	numeric	character varying(17)
1	45000	Profi
2	37000	Profi
3	50000	Fortgeschrittener
4	60000	Profi
5	55000	Fortgeschrittener
6	15000	Anfaenger
7	50000	Fortgeschrittener

vorher

personid	gehalt	erfahrung
integer	numeric	character varying(17)
1	45000	Profi
2	37000	Profi
3	75000	Fortgeschrittener
4	60000	Profi
5	55000	Fortgeschrittener
6	15000	Anfaenger
7	50000	Fortgeschrittener

nachher

```

class org.postgresql.Driver
update Mitarbeiter2 set gehalt=gehalt*1.5 where personid = 3
BUILD SUCCESSFUL (total time: 0 seconds)

```

SQL Injection

```
7 import java.sql.Connection;
8 import java.sql.DriverManager;
9 import java.sql.ResultSet;
10 import java.sql.ResultSetMetaData;
11 import java.sql.SQLException;
12 import java.sql.Statement;
13
14 public class JDBCParameterExample {
15
16     public static void outputResultSet(ResultSet rset) throws SQLException {...}
17
18
19
20
21
22
23
24
25
26
27
28     public static void main(String[] args) {
29         Connection conn = null;
30
31         try {
32             System.out.println(Class.forName("org.postgresql.Driver"));
33             conn = DriverManager.getConnection("jdbc:postgresql://localhost/Fotoagentur?user=postgres&password=.....");
34         } catch (Exception e) {
35             System.out.println("Fehler: " + e);
36             System.exit(-1);
37         }
38         if (conn != null) {
39             try {
40                 Statement sql_stmt = conn.createStatement();
41                 final String queryString = "update Mitarbeiter2 set gehalt=gehalt*1.5 where personid = " + args[0];
42                 System.out.println(queryString);
43                 boolean success = sql_stmt.execute(queryString);
44                 conn.close();
45             } catch (SQLException se) {
46                 System.out.println("Fehler: " + se);
47             }
48         }
49     }
50 }
```



args[0] = "3 OR gehalt<100000;"

personid	gehalt	erfahrung
integer	numeric	character varying(17)
1	45000	Profi
2	37000	Profi
3	75000	Fortgeschrittener
4	60000	Profi
5	55000	Fortgeschrittener
6	15000	Anfaenger
7	50000	Fortgeschrittener

personid	gehalt	erfahrung
integer	numeric	character varying(17)
1	67500	Profi
2	55500	Profi
3	11250	Fortgeschrittener
4	90000	Profi
5	82500	Fortgeschrittener
6	22500	Anfaenger
7	75000	Fortgeschrittener

vorher

nachher

```
class org.postgresql.Driver
update Mitarbeiter2 set gehalt=gehalt*1.5 where personid = 3 OR gehalt<100000;
BUILD SUCCESSFUL (total time: 0 seconds)
```

Prepared Statements

DEALLOCATE

updateMitarbeiter2;

Abfrage war erfolgreich nach 19 ms. Keine Zeilen geliefert.

PREPARE

updateMitarbeiter2 (int) AS

UPDATE

Mitarbeiter2 **SET** gehalt=gehalt*1.5

WHERE

personid=\$(1);

Abfrage war erfolgreich nach 13 ms. Keine Zeilen geliefert.

EXECUTE showFotograph(3);

id	name	vorname	geburtsdatum	gehalt	erfahrung
integer	character varying	character varying	date	numeric	integer
3	Mueller	Peter	1963-10-09	50000	2

showFotograph ist hier ein zweites prepared statement, das einfach den Inhalt der Tabelle Mitarbeiter2 für eine bestimmte personid ausgibt. Um das eigentlich update für personid=3 auszuführen, muss ich natürlich EXECUTE updateMitarbeiter2(3) aufrufen.

```
7 import java.sql.Connection;
8 import java.sql.DriverManager;
9 import java.sql.PreparedStatement;
10 import java.sql.ResultSet;
11 import java.sql.ResultSetMetaData;
12 import java.sql.SQLException;
13
14 public class JDBCPreparedStatementExample {
15
16     public static void outputResultSet(ResultSet rset) throws SQLException {...}
17
18     public static void main(String[] args) {
19         Connection conn = null;
20
21         try {
22             System.out.println(Class.forName("org.postgresql.Driver"));
23             conn = DriverManager.getConnection("jdbc:postgresql://localhost/Fotoagentur?user=postgres&password=.....");
24         } catch (Exception e) {
25             System.out.println("Fehler: " + e);
26             System.exit(-1);
27         }
28         if (conn != null) {
29             try {
30                 PreparedStatement preparedStatement = conn.prepareStatement("select * from FotografenKomplett2 where id = ?");
31                 preparedStatement.setInt(1, 6);
32                 ResultSet rset = preparedStatement.executeQuery();
33                 outputResultSet(rset);
34
35                 //irgendwann später dann:
36                 preparedStatement.setInt(1, 42);
37                 ResultSet rset2 = preparedStatement.executeQuery();
38                 outputResultSet(rset2);
39
40                 rset.close();
41                 conn.close();
42             } catch (SQLException se) {
43                 System.out.println("Fehler: " + se);
44             }
45         }
46     }
47 }
48
49 }
```

```
7 import java.sql.Connection;
8 import java.sql.DriverManager;
9 import java.sql.PreparedStatement;
10 import java.sql.SQLException;
11
12 public class JDBCPreparedStatementExample2 {
13
14     public static void main(String[] args) {
15         Connection conn = null;
16
17         try {
18             System.out.println(Class.forName("org.postgresql.Driver"));
19             conn = DriverManager.getConnection("jdbc:postgresql://localhost/Fotoagentur?user=postgres&password=.....");
20         } catch (Exception e) {
21             System.out.println("Fehler: " + e);
22             System.exit(-1);
23         }
24         if (conn != null) {
25             try {
26                 PreparedStatement preparedStatement = conn.prepareStatement("update Mitarbeiter2 set gehalt=gehalt*1.5 where personid = ?");
27
28                 preparedStatement.setInt(1, 6);
29                 preparedStatement.execute();
30
31                 //irgendwann später dann:
32                 preparedStatement.setInt(1, 42);
33                 preparedStatement.execute();
34
35                 conn.close();
36             } catch (SQLException se) {
37                 System.out.println("Fehler: " + se);
38             }
39         }
40     }
41 }
42 }
```

```
class org.postgresql.Driver
BUILD SUCCESSFUL (total time: 0 seconds)
```

```

7 import java.sql.Connection;
8 import java.sql.DriverManager;
9 import java.sql.PreparedStatement;
10 import java.sql.SQLException;
11
12 public class JDBCPreparedStatementExample2 {
13
14     public static void main(String[] args) {
15         Connection conn = null;
16
17         try {
18             System.out.println(Class.forName("org.postgresql.Driver"));
19             conn = DriverManager.getConnection("jdbc:postgresql://localhost/Fotoagentur?user=postgres&password=.....");
20         } catch (Exception e) {
21             System.out.println("Fehler: " + e);
22             System.exit(-1);
23         }
24         if (conn != null) {
25             try {
26                 PreparedStatement preparedStatement = conn.prepareStatement("update Mitarbeiter2 set gehalt=gehalt*1.5 where personid = ?");
27
28                 preparedStatement.setInt(1, 6);
29                 preparedStatement.execute();
30
31                 //irgendwann später dann:
32                 preparedStatement.setInt(1, 42);
33                 preparedStatement.execute();
34
35                 //irgendwann später dann:
36                 preparedStatement.setString(1, "3 OR gehalt<100000;");
37                 preparedStatement.execute();
38
39                 conn.close();
40             } catch (SQLException se) {
41                 System.out.println("Fehler: " + se);
42             }
43         }
44     }
45 }
46
47

```

```

class org.postgresql.Driver
Fehler: org.postgresql.util.PSQLException: ERROR: operator does not exist: integer = character varying
Hinweis: No operator matches the given name and argument type(s). You might need to add explicit type casts.
Position: 58
BUILD SUCCESSFUL (total time: 0 seconds)

```