

Java da ma'lumotlar ombori bilan ishlash

Reja:

1. **JDBC**
2. **Ma'lumotlar omboriga so'rovlarni amalga oshirish**
3. **PreparedStatement bilan ishlash**
4. **CallableStatement bilan ishlash**

Web application



Java



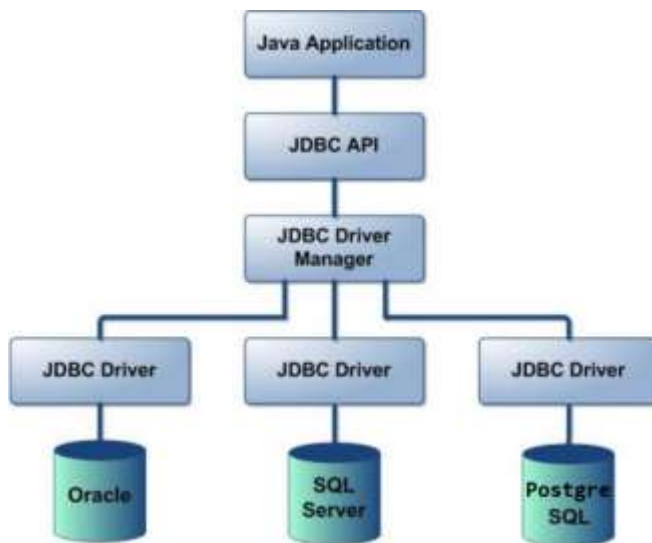
PostgreSQL

JDBC (Java DataBase Connectivity)

- Ma'lumotlar omboriga bog'lanish deb tarjima qilinadi.
- Java dasturni har qanday turdagi MOBT bilan qo'llash uchun ishlatiladi. Java.sql paketi orqali ishlatiladi
- Java dastur va MOBT o'rtasida bog'lovchi vosita sifatida xizmat qiladi
- Har bir MOBT ning o'zining drayveri mavjud. Qaysi MOBT ga ulamoqchi bo'lsangiz o'sha drayverni yuklaysiz xolos

JDBC

➤ JDBC – ma'lumotlar omboriga maxsus URL orqali bog'lanishni ta'minlovchi DRAYERlar konsepsiyasiga asoslangan. Drayverlar esa dastur ishlayotgan paytda dinamik tarzda yuklanishi mumkin.



JDBC afzalliklari

- Foydalanish uchun juda qulay: foydalanuvchi(dasturchi) foydalanmoqchi bo'lgan ma'lumotlar omborini boshqarish tizimi(MOBT) bilan ishlash ko'nikmasiga ega bo'lmasdan turib ham bimalol dastur tuza oladi;
- Agarda kompaniya boshqa MOBT ga o'tishni xoxlasa kod deyarli o'zgarmaydi. O'zgarishlar esa aynan SQL dialektlari bilan bog'liq bo'ladi;
- Hech qanday client dasturni o'rnatishga hojat yo'q;
- Istalgan MOBTga yengil yoziladigan URL orqali bo'g'lanish mumkin;

JDBC API interfacelari

Asosan 2 xil turdagi interfacega ega: birinchisi – dastur yaratuvchilar uchun va ikkinchisi driver yaratuvchilar uchun mo'ljallangan.

- MOBT bilan bo'g'lanish ***java.sql.Connection*** interface ini amalga oshirgan klass orqali amalga oshiriladi.
- Bo'g'lanishga ega bo'lgan holda Statement obe'kti orqali MOBT ga SQL so'rovlarni amalga oshirish mumkin. Statement ni quyidagi turlari mavjud:

java.sql.Statement – umumiy foydalanishga mo'ljallangan Statement

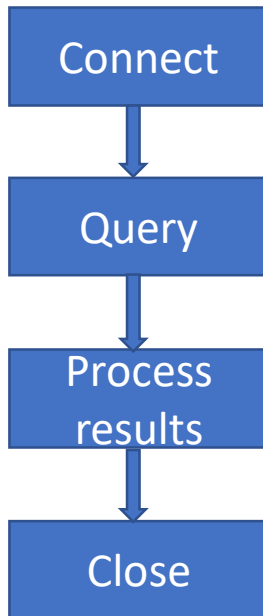
java.sql.PreparedStatement – parametrlarini berish imkoni mavjud bo'lgan so'rovlarni bajarish uchun mo'ljallangan Statement

java.sql.CallableStatement – MO funksiya va proseduralarini chaqirish uchun mo'ljallangan Statement

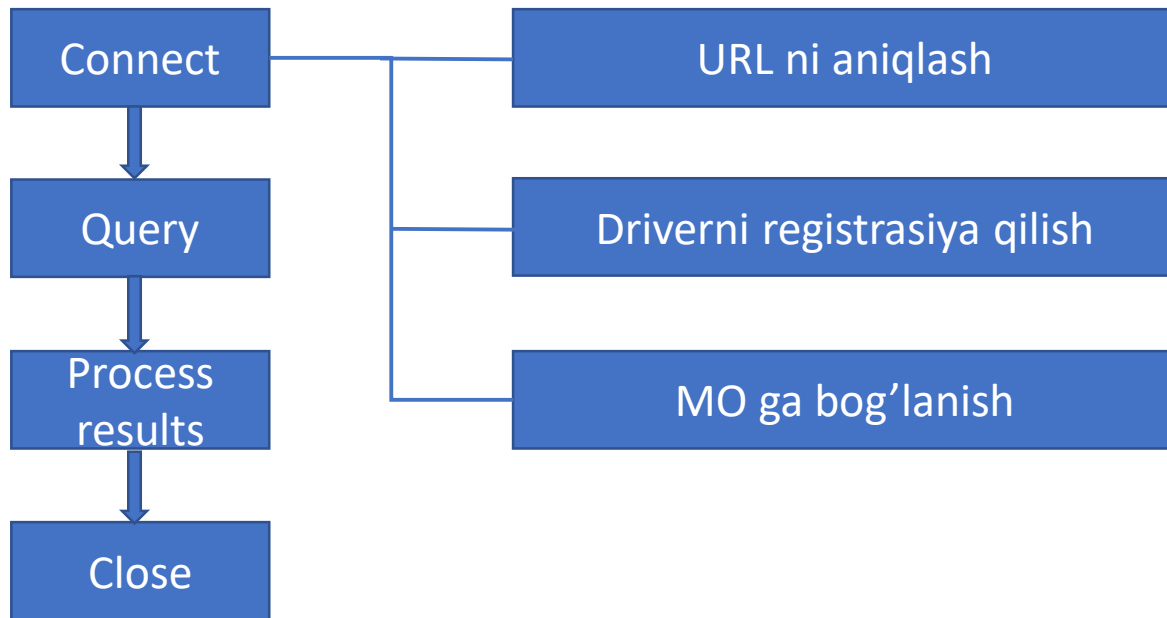
- ***java.sql.ResultSet*** – so'rov natijasi bilan ishlashga mo'ljallangan interface

JDBC orqali MO ga so'rovlarni amalga oshirish

JDBC orqali MO ga so'rovlarni amalga oshirish quyidagi bosqichlarda amalga oshiriladi:



1-bosqich: Connect



URLni aniqlash

- URL="jdbc:" + mobt + "://" + host + ":" + port + "/" + DB;
- *mobt* = "postgresql";
- *host* = "localhost";
- *port* = "5432";
- *db* = "m";

Driverni registrasiya qilish

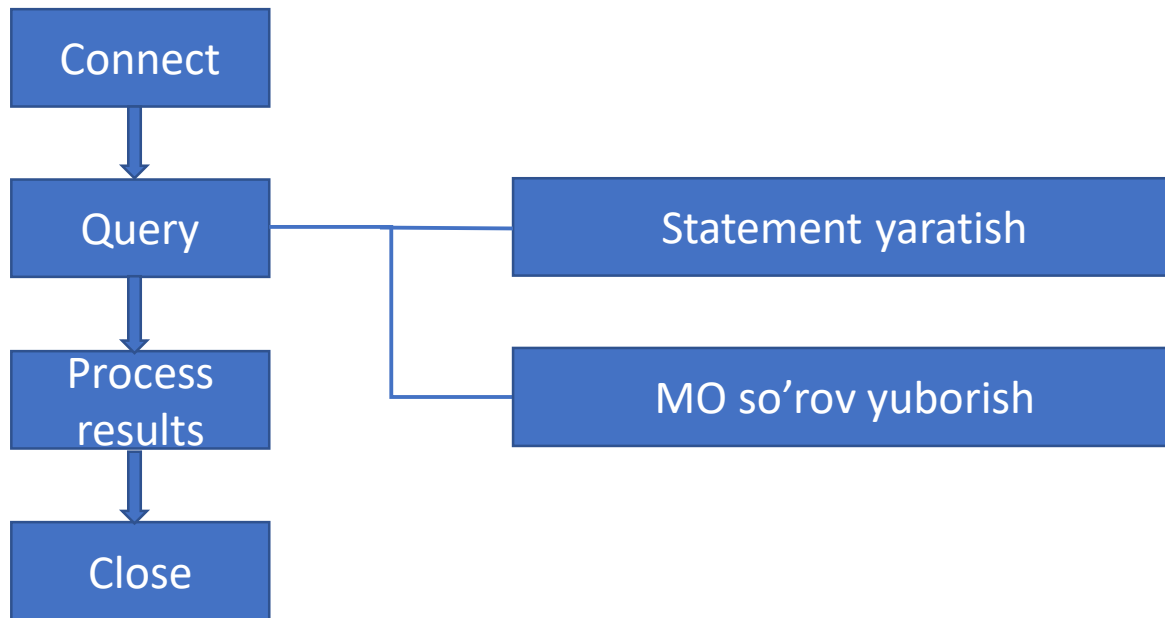
```
Class.forName("org.postgresql.Driver  
r")
```

MO ga bog'lanish

```
dbUser = "postgres";  
dbPassword = "*****";
```

```
Connection connection =  
    DriverManager.getConnection(url, dbUser,  
dbPassword);
```

2-bosqich: Query



Statement yaratish

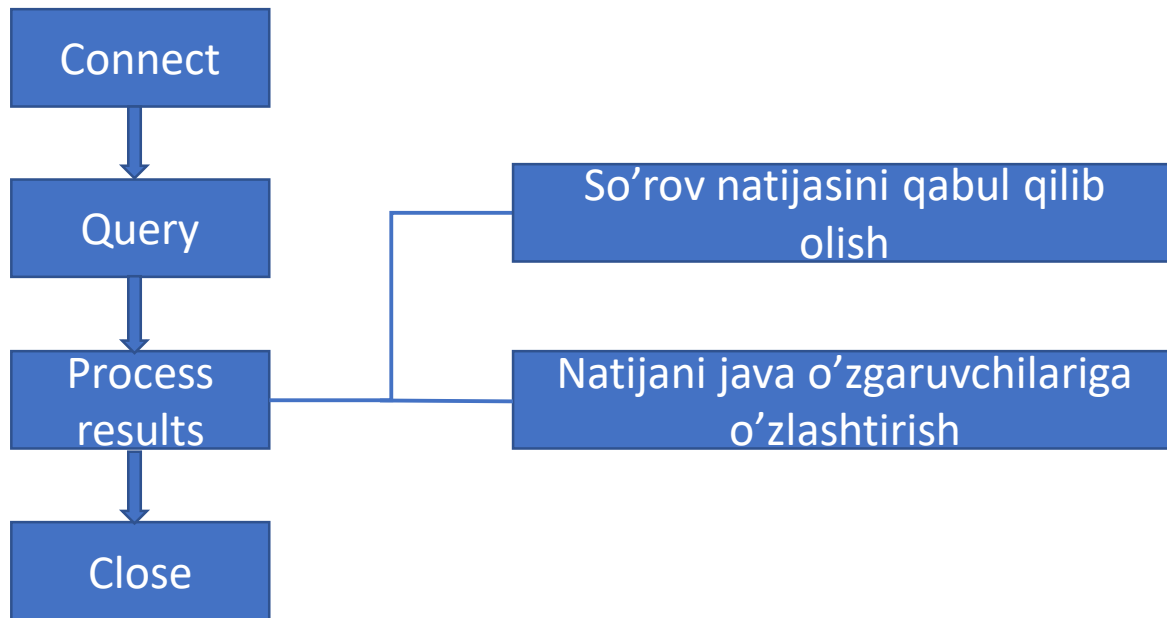
```
Statement statement =  
connection.createStatement();
```

MO so'rov yuborish

```
String sql = "select * from users;";
```

```
statement.executeQuery(s  
ql);
```


2-bosqich: Query



So'rov natijasini qabul qilib olish

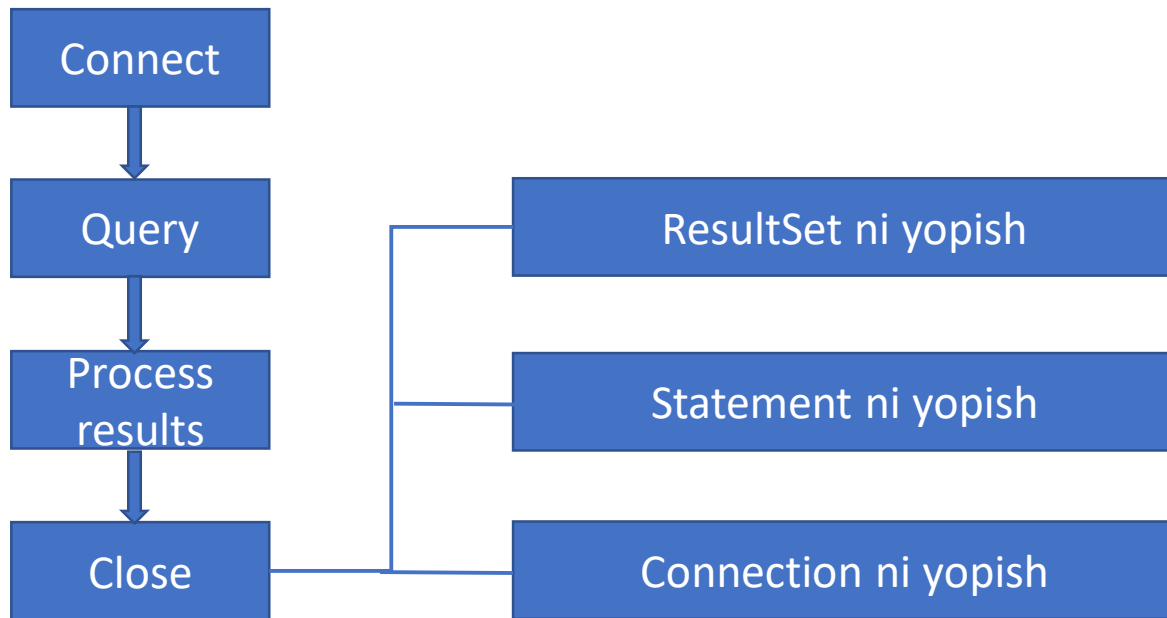
```
String sql = "select * from users;";
```

```
ResultSet resultSet = statement.executeQuery(sql);
```

Natijani java o'zgaruvchilariga o'zlashtirish

```
while (resultSet.next()) {  
    System.out.print("id=" + resultSet.getInt(1));  
    System.out.print("\nname=" + resultSet.getString("name"));  
    System.out.print("\nusername=" + resultSet.getString(3));  
    System.out.println("\tpassword=" + resultSet.getString(4));  
    System.out.println("\temail=" + resultSet.getString(5));  
    System.out.println("\tphone=" + resultSet.getString(4));  
}
```

2-bosqich: Query



PreparedStatement bilan ishlash

```
String sql = "select * from users where name like `%?%`?  
              and ? Email like `%?`";
```

```
PreparedStatement preparedStatement = connection.prepareStatement(sql);  
preparedStatement.setString(1, "ali");  
preparedStatement.setString(2, "gmail.com");  
ResultSet resultSet = preparedStatement.executeQuery();
```

CallableStatement bilan ishlash

```
String sql = "{call add_user(?,?,?,?,?,?,?)}";
```

```
CallableStatement callableStatement = connection.prepareCall(sql);  
callableStatement.setString(1, "Solijon");  
callableStatement.setString(2, "solijon");  
callableStatement.setString(3, "123");  
callableStatement.setString(4, "solijon@gmail.com");  
callableStatement.setString(5, "+998997776655");  
callableStatement.registerOutParameter(6, 16);  
callableStatement.registerOutParameter(7, Types.VARCHAR);  
callableStatement.executeUpdate();
```

E'tiboringiz uchun raxmat