SUNBEAM

Java EE™

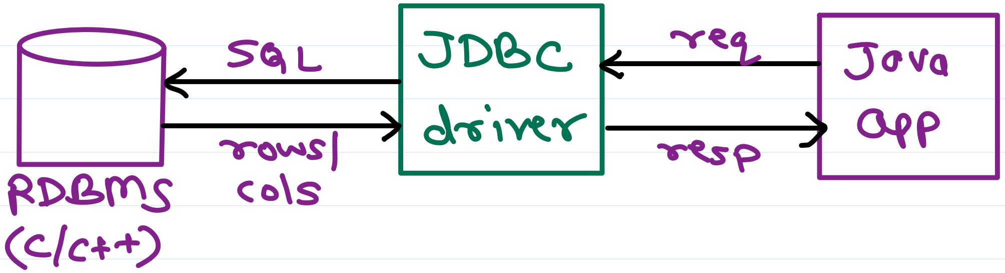
Advanced Java

Trainer: Nilesh Ghule

JDBC

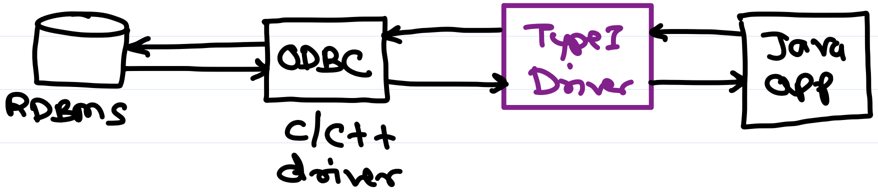
* specification by Sun Microsystem/Oracle.

* Standard way to connect RDBMS from java appln.

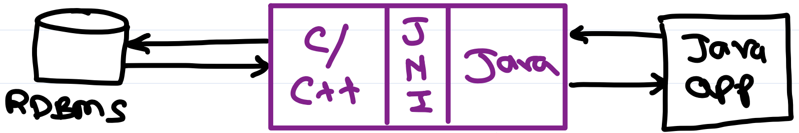


* Types of JDBC drivers

① Type I : JDBC ODBC Bridge



② Type II : Partial C & Java driver



Type II driver

* JDBC std is given in form of intfs & helper classes → java.sql package.

① Driver → connect to db (returns Connection)

② Connection → tx mgmt, create stmt

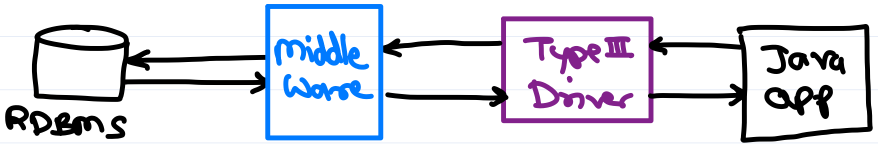
③ Statement → execute sql & get result

PreparedStatement → execute parameterized query & get the result

CallableStatement → execute stored procedure & get the result

④ ResultSet → result of select query (rows & cols)

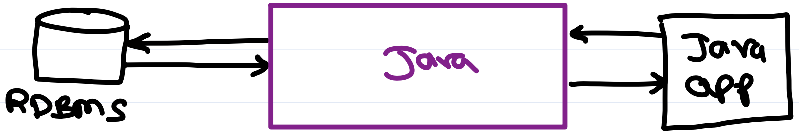
③ Type III : middleware driver



① DriverManager

② Date, Timestamp, Blob, ... (types)

④ Type IV : True Java driver



Type IV driver

JDBC programming steps

Statement

⓪ add jdbc driver (jar) in project classpath.

Project → Properties → Java Build Path → Libraries (ClassPath) → Add External Jar → select driver jar → Apply / ok.

① load & register driver class.

```
Class.forName("driver class name");
```

→ com.mysql.cj.jdbc.Driver.

② create jdbc connection.

```
con = DriverManager.getConnection(dbUrl, dbUser, dbPasswd);
```

jdbc:mysql://server:3306/db

③ create sql statement.

```
stmt = con.createStatement();
```

④ execute sql query & process result.

```
cnt = stmt.executeUpdate("sql");
```

↑ int = num of rows affected

↑ non-select

```
rs = stmt.executeQuery("select sql");
while(rs.next()) {
    v1 = rs.getInt("col1");
    v2 = rs.getString("col2");
    v3 = rs.getDate("col3");
    v4 = rs.getDouble("col4");
    ...
}
rs.close();
```

⑤ close stmt & connection.

```
stmt.close();
con.close();
```

JDBC programming steps

PreparedStatement

⓪ add jdbc driver (jar) in project classpath.

Project → Properties → Java Build Path → Libraries (ClassPath) → Add External Jar → select driver jar → Apply / ok.

① load & register driver class.

```
Class.forName("driver class name");
```

→ com.mysql.cj.jdbc.Driver.

② create jdbc connection.

```
con = DriverManager.getConnection(dbUrl, dbUser, dbPasswd);
```

jdbc:mysql://server:3306/db

③ create sql statement.

```
stmt = con.prepareStatement(sql);
```

select or non-select query with params (?).

e.g. DELETE FROM emp WHERE sal > ?

SELECT * FROM emp WHERE sal BETWEEN ? AND ?

④ execute sql query & process result.

```
stmt.setInt(1, v1);
stmt.setString(2, v2);
...
```

} set param values.

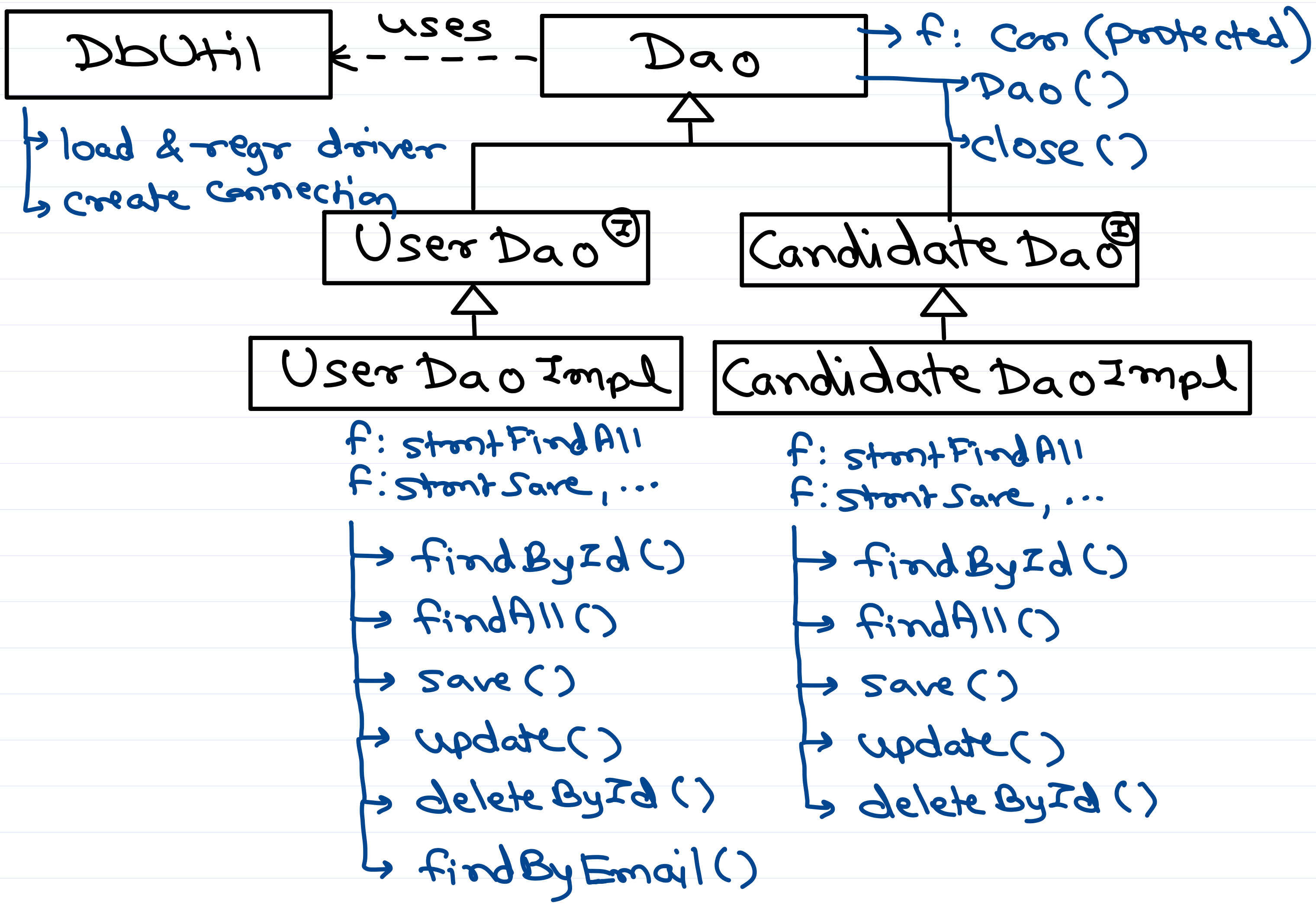
```
cnt = stmt.executeUpdate(x);
```

```
rs = stmt.executeQuery(x);
while(rs.next()) {
    v1 = rs.getInt("col1");
    v2 = rs.getString("col2");
    v3 = rs.getDate("col3");
    v4 = rs.getDouble("col4");
    ...
}
rs.close();
```

⑤ close stmt & connection.

```
stmt.close();
con.close();
```

JDBC DAO



Dao
- f: con (protected)
- Dao()
- close()

DbUtil
- load & regr driver
- create connection

UserDaoImpl
- f: stmtFindAll
- f: stmtSave, ...
- findById()
- findAll()
- save()
- update()
- deleteById()
- findByEmail()

CandidateDaoImpl
- f: stmtFindAll
- f: stmtSave, ...
- findById()
- findAll()
- save()
- update()
- deleteById()

SUNBEAM

Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>