Работа с базами данных

JDBC

Валерий Алексеевич Овчинников
valery.ovchinnikov@phystech.edu

Мотивация

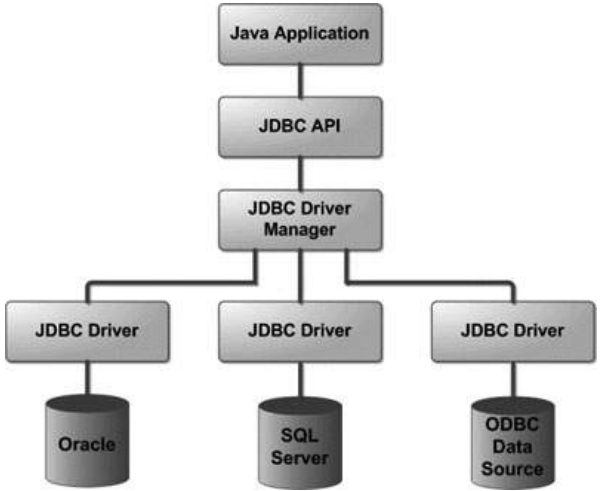
- Практически любое серьезное приложение что-то хранит (данные, конфигурацию, состояние)

- Практически любое серьезное приложение что-то хранит (данные, конфигурацию, состояние)
- Самым распространенным способом хранить данные является использование реляционных баз данных

- Практически любое серьезное приложение что-то хранит (данные, конфигурацию, состояние)
- Самым распространенным способом хранить данные является использование реляционных баз данных
- При работе с RDBMS из Java прямо или опосредованно исползуется JDBC

JDBC

JDBC: Architecture



JDBC: Driver

Существует 4 типа JDBC драйверов:

1. JDBC-ODBC мост (является адаптером к ODBC драйверу)
2. JDBC - Native API (использует нативные вызовы в базу)
3. JDBC-Net pure Java (использует сервер-посредник)
4. 100% pure Java (предпочтительный)

JDBC: Driver

Нужно узнать имя драйвера для вашей RDBMS

RDBMS	Driver name
MySQL	`com.mysql.jdbc.Driver`
Oracle	`oracle.jdbc.driver.OracleDriver`
DB2	`COM.ibm.db2.jdbc.net.DB2Driver`
Sybase	`com.sybase.jdbc.SybDriver`
PostgreSQL	`org.postgresql.Driver`

JDBC: Driver

Драйвер нужно зарегестрировать (достаточно загрузить класс)

```
Class.forName(DB_DRIVER);
```

JDBC: Connection

Для подключения к базе нужно составить jdbc-url

RDBMS	URL format
MySQL	`jdbc:mysql://hostname/database`
Oracle	`jdbc:oracle:thin:@hostname:port:databaseName`
DB2	`jdbc:db2:hostname:port/databaseName`
Sybase	`jdbc:sybase:Tds:hostname:port/database`
PostgreSQL	`jdbc:postgresql://host:port/database`

JDBC: Connection

```
try (Connection conn = DriverManager.getConnection(String url)) {}
Connection conn = DriverManager
        .getConnection(String url, Properties prop);
Connection conn = DriverManager
        .getConnection(String url, String user, String password);
```

JDBC: Statements

Существует 3 типа Statement:

1. Statement – для выполнения статических SQL выражений
2. PreparedStatement – для SQL выражений с параметрами (рекомендуются, переиспользуемые)
3. CallableStatement – для вызова хранимых (в базе) процедур

JDBC: Statement

```
try (Statement st = conn.createStatement()) {
  boolean st.execute(String sql);
  int st.executeUpdate(String sql); // insert, update, delete
  ResultSet st.executeQuery(String sql); // select
}
```

JDBC: PreparedStatement

```
try (PreparedStatement st = conn.prepareStatement(sql)) {
  boolean st.execute(String sql);
  int st.executeUpdate(String sql);
  ResultSet st.executeQuery(String sql);
}
```

JDBC: Transactions

По умолчанию включен autocommit

```
conn.setAutoCommit(false);
conn.commit();
conn.rollback();
```

JDBC: Transactions

```
Savepoint savepoint = conn.setSavepoint("Savepoint1");
try {
    stmt.executeUpdate(SQL1);
    stmt.executeUpdate(SQL2);
} catch(SQLException e) {
  conn.rollbacl(savepoint);
}
```

JDBC: Batch updates

Если нужно выполнить несколько команд, то эффективнее использовать batching

```
conn.setAutoCommit(false);
st.addBatch(sql1);
st.addBatch(sql2);
st.executeBatch();
```

JDBC: JOOQ

Существуют обертки над JDBC (e.g. Spring JDBCTemplated, JOOQ)
JOOQ:

```
DSLContext create = DSL.using(conn, SQLDialect.POSTGRES);
Result<Record> result = create.select().from(AUTHOR).fetch();
for (Record r : result) {
    Integer id = r.getValue(AUTHOR.ID);
    String firstName = r.getValue(AUTHOR.FIRST_NAME);
    String lastName = r.getValue(AUTHOR.LAST_NAME);

    System.out.println("ID: " + id + " first name: " +
        firstName + " last name: " + lastName);
}
```

Revolut МФТИ