JDBC – программный пакет для работы с БД

Основные пакеты в Java для работы с БД

- `java.sql.*` - основные классы для работы с данными
- `sun.jdbc.odbc.*` - классы и интерфейсы моста JDBC – ODBC.

Основные классы и интерфейсы

- `java.sql.Connection` – соединение с драйвером БД;
- `java.sql.DriverManager` – загрузка и манипулирование драйверами БД;
- `java.sql.Statement` – объекты для исполнения SQL-предложений;
- `java.sql.ResultSet` – объекты для обработки результатов `Select`-запросов;
- `java.sql.SQLException` – прерывания при работе с БД;
- `sun.jdbc.odbc.JdbcOdbcDriver` – драйвер моста JDBC – ODBC.

JDBC – программный пакет для работы с БД

Схема работы с БД из программ на Java

1. Загрузить класс(ы), реализующие необходимые драйверы
   ```
   Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
   ```
2. Установить соединение с БД, используя загруженный драйвер
   ```
   DriverManager.getConnection("jdbc:odbc:dsn");
   ```
3. Создать объект(ы) для исполнения SQL-команд
   ```
   connection.createStatement();
   ```
4. Исполнять необходимые SQL-команды
   ```
   stmt.executeUpdate("Delete From MyTable Where Id=1");
   stmt.executeQuery("Select * From MyTable");
   ```
5. Обрабатывать полученные таблицы
   ```
   result.getString("fieldName");
   ```
6. Закрыть открытые соединения
   ```
   connection.close();
   ```

JDBC – программный пакет для работы с БД

Пример программы

```
import java.sql.*;

public static void main(String[] args) {
  try {
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    Connection conn = DriverManager.getConnection("jdbc:odbc:dsn");
    Statement stmt = conn.createStatement();
    ResultSet rs = stmt.executeQuery("Select * From positions");
    while (rs.next()) {
      String s = rs.getString("name");
      System.out.println("Position name = " + s);
    }
  } catch (SQLException ex) {
    ex.printStackTrace();
  }
  try { conn.close(); } catch (SQLException e) {}
}
```

JDBC – программный пакет для работы с БД

Загрузка драйвера и установление соединения с БД

1. При загрузке класса драйвера он регистрируется в `DriverManager`.

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
com.mysql.jdbc.Driver.class;
import com.microsoft.jdbc.sqlserver.*;
Driver driver = new SQLServerDriver();
```

2. Установить соединение с БД можно, используя менеджер

```
Connection conn =
           DriverManager.getConnection("jdbc:odbc:dsn");
Connection conn = DriverManager.getConnection(
      "jdbc:mysql://localhost:3306/test","root","root");
Properties props = new Properties();
props.put("user", "admin");
props.put("password", "myPwd");
Connection conn = driver.connect(
   "jdbc:microsoft:sqlserver://localhost:1433" , props);
```

JDBC – программный пакет для работы с БД

Интерфейс Connection

1. Создание «предложений» для работы с БД
```
Statement stmt = connection.createStatement();
PreparedStatement pst = connection.prepareStatement(sql);
CallableStatement cst = connection.prepareCall(sql);
```
2. Работа с транзакциями
```
connection.setAutoCommit(false);
connection.commit();
connection.rollback();
```
3. Закрытие соединения
```
connection.close();
```

JDBC – программный пакет для работы с БД

Создаем в базе данных таблицы и заносим данные

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
try {
  Connection conn = DriverManager.getConnection(
                    "jdbc:odbc:projects");
  Statement stmt = conn.createStatement();
  String createTable = "CREATE TABLE roles (“ +
    "Sortorder INTEGER, Name VARCHAR, PRIMARY KEY (Sortorder))“;
  stmt.executeUpdate(createTable);
  String insertData1 = "INSERT INTO roles (Sortorder, Name) " +
    "VALUES(10, 'Project Leader')“;
  String insertData2 = "INSERT INTO roles (Sortorder, Name) " +
    "VALUES(20, 'Developer')";
  stmt.executeUpdate(insertData1);
  stmt.executeUpdate(insertData2);
  conn.close();
} catch (SQLException x) {
  x.printStackTrace();
}
```

JDBC – программный пакет для работы с БД

Параметризованные SQL-предложения

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
try {
  Connection conn = DriverManager.getConnection(
                    "jdbc:odbc:projects");
  String insertData = "INSERT INTO roles (Sortorder, Name) " +
    "VALUES(?, ?)";
  PreparedStatement stmt = conn.prepareStatement(insertData);
  stmt.setInt(1, 30);
  stmt.setString (2, "QA Engineer");
  stmt.executeUpdate();
  conn.close();
} catch (SQLException x) {
  x.printStackTrace();
}
```

JDBC – программный пакет для работы с БД

Получение данных с помощью SELECT-предложений

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
try {
  Connection conn = DriverManager.getConnection(
                    "jdbc:odbc:projects");
  String selectData = "SELECT Name, Salary FROM positions";
  Statement stmt = conn.createStatement(selectData);
  ResultSet rs = stmt.executeQuery();
  System.out.println("Name\tSalary");
  System.out.println("----------------------");
  while (rs.next()) {
    String name = rs.getString("Name");
    int salary = rs.getInt("Salary");
    System.out.println(name + '\t'
  }
  conn.close();
} catch (SQLException x) {
  x.printStackTrace();
}
```

JDBC – программный пакет для работы с БД

Исследование полученных данных

```
private static void printTable(Connection conn, String sql) {
  try {
    Statement stmt = conn.createStatement();
    ResultSet rs = stmt.executeQuery(sql);
    ResultSetMetaData rsmd = rs.getMetaData();
    int colCount = rsmd.getColumnCount();
    for (int i = 1; i <= colCount; ++i) {
      System.out.print(rsmd.getColumnName(i) + '\t');
    }
    System.out.println(); System.out.println("--------------");
    while(rs.next()) {
      for (int i = 1; i <= colCount; ++i) {
        System.out.print("" + rs.getObject(i) + '\t');
      }
      System.out.println();
    }
  } catch(SQLException x) {
    System.out.println("Couldn't print table: " + x.getMessage());
  }
}
```

JDBC – программный пакет для работы с БД

Какую еще информацию можно получить, исследуя метаданные?

```
ResultSetMetaData rsmd = rs.getMetaData();
int count = rsmd.getColumnCount();
String fieldName = rsmd.getColumnName(i);
int type = rsmd.getColumnType(i);
    // Types.VARCHAR, Types.INTEGER,...
String typeName = rsmd.getColumnTypeName(i);
    // Как его возвращает драйвер БД, например "counter".
String className = rsmd.getColumnClassName(i);
    // С точки зрения языка Java, например, "java.math.BigDecimal".
boolean curr = rsmd.isCurrency(i);
int nullable = rsmd.isNullable(i);
    // columnNullable, columnNoNulls, columnNullableUnknown
```

Можно также получать метаданные о таблице, самой базе данных,…

```
DatabaseMetaData md = conn.getMetaData();
int maxRowSize = md.getMaxRowSize();
ResultSet metaRs = md.getTables(catalog, schema, tabName, types);
```

JDBC – программный пакет для работы с БД

Работа с курсором: изменение, добавление и удаление данных

```
// Увеличиваем все зарплаты на 10 процентов
private static void updateSalaries(Connection conn) {
  try {
    Statement stmt = conn.createStatement(
        ResultSet.TYPE_SCROLL_SENSITIVE, // Можно перемещать
        ResultSet.CONCUR_UPDATABLE);     // Можно изменять данные
    ResultSet rs = stmt.executeQuery("Select * From positions");
    while (rs.next()) {
      rs.updateInt("Salary", (int)(rs.getInt("Salary") * 1.1));
      rs.updateRow();
    }
  } catch (SQLException x) {
    x.printStackTrace();
  }
}

// Конечно, того же эффекта можно было бы добиться,
// просто исполнив предложение
// UPDATE positions SET Salary = Salary * 1.1
```

JDBC – программный пакет для работы с БД

Если курсор перемещаемый (`TYPE_SCROLL_SENSITIVE`),
то с ним можно выполнять следующие действия:

```
ResultSet rs = ...;
rs.absolute(1);
rs.beforeFirst();
rs.afterLast();
rs.first();
rs.last();
rs.next()
rs.previous();
rs.relative(-5);
```

Если курсор перемещаемый только вперед (`TYPE_FORWARD_ONLY`),
то с ним можно выполнять только `next()`

JDBC – программный пакет для работы с БД

Если курсор «заменяемый» (`TYPE_CONCUR_UPDATEABLE`),
то можно заменять, добавлять и удалять данные:

```
// Замена
rs.updateInt(column, newData);
... // другие изменения в текущей записи
rs.updateRow();

// Добавление
rs.updateString(column, newData);
... // другие добавления в запись
rs.insertRow();

// Удаление
rs.deleteRow();
```

Если `SELECT`-предложение сложное, то может быть, данные невозможно будет изменить, даже несмотря на то, что курсор `UPDATEABLE`.

JDBC – программный пакет для работы с БД

Подготовка процедуры, составленной из нескольких подряд исполняемых SQL-предложений (batch).

```
Statement stmt = conn.createStatement();
stmt.addBatch("Update positions Set Salary = Salary * 1.2 " +
              "Where Name Like '%Engineer%'");
stmt.addBatch("Update positions Set Salary = Salary * 1.5 " +
              "Where Name Like '%Director%'");
int[] results = stmt.executeBatch();
stmt.clearBatch();
```