

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

Старший преподаватель

должность, уч. степень, звание

подпись, дата

Н.В. Путилова

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №8

Проектирование взаимодействия базы данных и приложения

по курсу: Проектирование баз данных

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

4932

подпись, дата

Н.С. Иванов

инициалы, фамилия

Санкт-Петербург 2021

Текст задания

8. костюмерная театра: роль, спектакль, название костюма, деталь костюма, размер, автор модели, дата разработки

а. спектакли, в которых используются костюмы, имеющие в названии слово «принц»

б. костюм, в котором есть и плащ и штаны

в. спектакль, на который пока нет костюмов

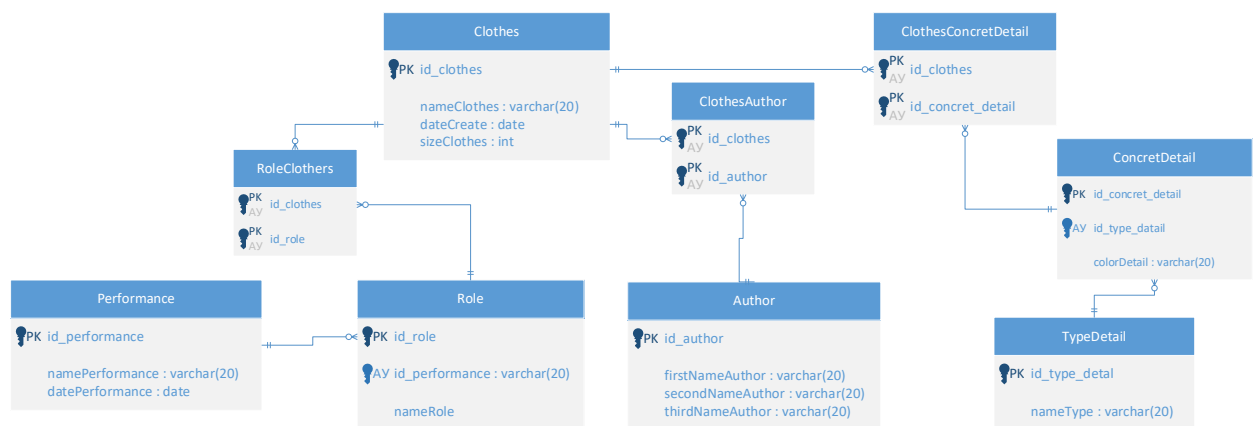
г. роль, к которой разрабатывался самый старый из костюмов

д. автор, разработавший наибольшее число костюмов

е. костюм, в котором есть все типы деталей

ж. автор, не разрабатывавший костюмы к «Золушке», но разрабатывавший к «Мастеру и Маргарите»

Физическая модель



Метод «С подключением источника данных» (Замена)

```
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.sql.*;
import java.util.Vector;

public class BindindMethod {
    // JDBC URL, username and password of MySQL server
    private static final String url =
        "jdbc:mysql://localhost:3306/database_design_course";
    private static final String user = "root_user";
    private static final String password = "root_user";

    public static void main(String[] args) throws SQLException {
        String query = "select * from author";

        // opening database connection to MySQL server
        Connection con = DriverManager.getConnection(url, user, password);
        // getting Statement object to execute query
        Statement stmt = con.createStatement();
        // executing SELECT query
        ResultSet rs = stmt.executeQuery(query);

        // It creates and displays the table
        JTable table = new JTable(buildTableModel(rs));
        JOptionPane.showMessageDialog(null, new JScrollPane(table));

        con.close();
        stmt.close();
        rs.close();
    }

    public static DefaultTableModel buildTableModel(ResultSet rs)
        throws SQLException {

        ResultSetMetaData metaData = rs.getMetaData();

        // names of columns
        Vector<String> columnNames = new Vector<String>();
        int columnCount = metaData.getColumnCount();
        for (int column = 1; column <= columnCount; column++) {
            columnNames.add(metaData.getColumnName(column));
        }

        // data of the table
        Vector<Vector<Object>> data = new Vector<Vector<Object>>();
        while (rs.next()) {
            Vector<Object> vector = new Vector<Object>();
            for (int columnIndex = 1; columnIndex <= columnCount;
columnIndex++) {
                vector.add(rs.getObject(columnIndex));
            }
            data.add(vector);
        }

        return new DefaultTableModel(data, columnNames);
    }
}
```

Метод «Цикл»

```
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
```

```

import java.sql.*;
import java.util.Vector;

/**
 * Simple Java program to connect to MySQL database running on localhost and
 * running SELECT and INSERT query to retrieve and add data.
 * @author Javin Paul
 */
public class LoopMethod {
    // JDBC URL, username and password of MySQL server
    private static final String url =
"jdbc:mysql://localhost:3306/database_design_course";
    private static final String user = "root_user";
    private static final String password = "root_user";

    public static void main(String[] args) throws SQLException {
        String query = "select * from author";

        // opening database connection to MySQL server
        Connection con = DriverManager.getConnection(url, user, password);
        // getting Statement object to execute query
        Statement stmt = con.createStatement();
        // executing SELECT query
        ResultSet rs = stmt.executeQuery(query);

        loopMethod(rs);

        con.close();
        stmt.close();
        rs.close();
    }

    public static void loopMethod(ResultSet rs) throws SQLException {
        Vector<Vector<String>> ownerList = new Vector<>();
        var i = 0;
        while (rs.next()) {
            Vector<String> a = new Vector<String>(4);
            a.add(Integer.toString(rs.getInt("id_author")));
            a.add(rs.getString("firstNameAuthor"));
            a.add(rs.getString("secondNameAuthor"));
            a.add(rs.getString("thirdNameAuthor"));
            ownerList.add(a);
        }
        rs.close();

        // Column Names
        Vector<String> columnNames = new Vector<String>(4);
        columnNames.add("id");
        columnNames.add("Фамилия");
        columnNames.add("Имя");
        columnNames.add("Отчество");

        JTable table = new JTable(new DefaultTableModel(ownerList,
columnNames));
        JOptionPane.showMessageDialog(null, new JScrollPane(table));
    }
}

```

Метод «Hibernate»

Класс

```

import lombok.Getter;
import lombok.NoArgsConstructor;

```

```

import lombok.Setter;

import javax.persistence.*;
import java.io.Serializable;
import java.io.Serializable;

@Entity
@NoArgsConstructor
@Table(name = "author", schema = "database_design_course")
public class Author implements Serializable {
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Getter @Setter protected Long id_author;
    @Getter @Setter protected String firstNameAuthor;
    @Getter @Setter protected String secondNameAuthor;
    @Getter @Setter protected String thirdNameAuthor;
}

```

Конфиг

```

<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD//EN"
    "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
    <session-factory>

        <property
name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
        <property
name="connection.url">jdbc:mysql://localhost:3306/database_design_course</pro
perty>
        <property
name="connection.driver_class">com.mysql.jdbc.Driver</property>

        <property name="connection.username">root_user</property>
        <property name="connection.password">root_user</property>

        <property name="hbm2ddl.auto">update</property>

        <mapping class="Author"/>

    </session-factory>
</hibernate-configuration>

```

Код

```

import org.hibernate.Session;

import javax.persistence.Query;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.sql.*;
import java.util.List;
import java.util.Vector;

public class HibernateMethod {

    public static void main(String[] args) throws SQLException {
        // It creates and displays the table
        JTable table = new JTable(getData());
        JOptionPane.showMessageDialog(null, new JScrollPane(table));
    }

    public static DefaultTableModel getData() {

```

```

List<Author> rs = getItems();

// Column Names
Vector<String> columnNames = new Vector<String>(4);
columnNames.add("id");
columnNames.add("Фамилия");
columnNames.add("Имя");
columnNames.add("Отчество");

Vector<Vector<String>> data = new Vector<>();
for (var r : rs ) {
    Vector<String> a = new Vector<String>(4);
    a.add(Long.toString(r.getId_author()));
    a.add(r.getFirstNameAuthor());
    a.add(r.getSecondNameAuthor());
    a.add(r.getThirdNameAuthor());
    data.add(a);
}

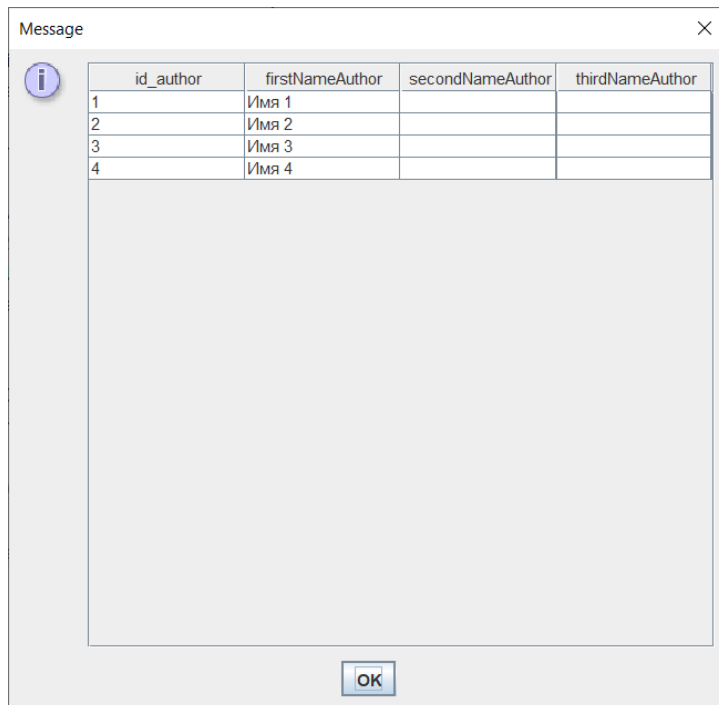
return new DefaultTableModel(data, columnNames);
}

public static List<Author> getItems(){
    Session session = HibernateSessionFactory.createSession();
    Query query = session.createQuery("from Author"); // select * from
authors
    var result = query.getResultList();
    session.close();
    return result;
}
}

```


Демонстрация

1

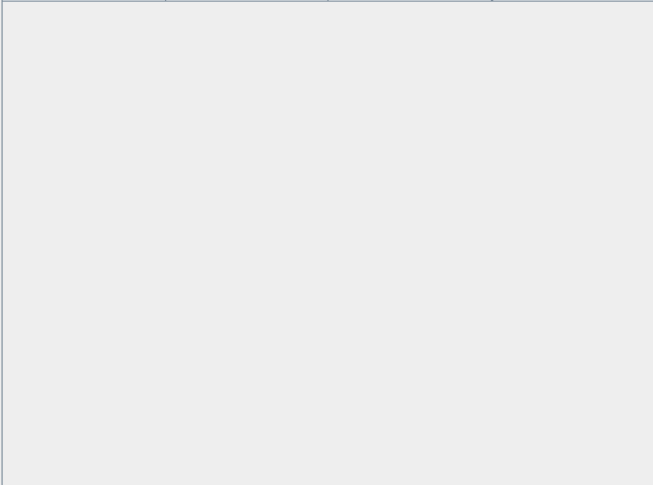



2

Message ×




id	Фамилия	Имя	Отчество
1	Имя 1		
2	Имя 2		
3	Имя 3		
4	Имя 4		





3

Message ×



id	Фамилия	Имя	Отчество
1	Имя 1		
2	Имя 2		
3	Имя 3		
4	Имя 4		

