

جافا بیانات قاعدة اتصال سایق استخدام

لتأسیس الخطوات من سلسلة اتباع علیک، في ۱۰۰۰ سایق لاستخدام واضحه بطريقه بخطوه خطوه العمليه خلال سأق ودك أدناه، الجناتي. مع وتعامل استعلامات، تنفيذ بيانت، قاعدة إلى اتصال وممارسات.

مشروعك إلى ۱۰۰۰ سایق إضافة: ۱ الخطوه

الخطوه ۱: مثلم معها تعامل الـ بيانت لقاعدة ۱۰۰۰ المنسوب لـ سایق إلى تحتاج أولًا، الخطوه. سایقه بيانت قاعدة مزود كل يوفر

لمي المثال، سبيل علی pom.xml ملف إلى التبعية أضاف: ۱۰۰۰ لمشاريع:

```
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.33</version> <!-- -->
</dependency>
```

لمي المثل:

```
<dependency>
    <groupId>org.postgresql</groupId>
    <artifactId>postgresql</artifactId>
    <version>42.7.3</version> <!-- -->
</dependency>
```

من ۱۰۰۰ مثلم سایق مزود موقع من ۱۰۰۰ سایق ۱۰۰۰ ملف بتزنيل قم: غير لمشاريع ۱۰۰۰ يدوياً. مشروعك تصنيف مسار إلى وأضافه ۱۰۰۰.۱۰۰۰

۱۰۰۰ سایق تحميـل: ۲ الخطوه

مع Class.forName(). بـ استخدام صريح بشكل سـايـق تحـميـل عـلـيـك كان، ۴.۰ قـبـل الـ قـديـمة ۴.۰ إـصـدارـات فـي الخـدمـة. مـزوـد آـلـيـة عـبـر تـلـقـائـيـاً سـايـق تـسـجـيـل يـتـم حـيـث اـخـتـيـارـيـة الـ خـطـوـة هـذـه، ۴.۰ الـ حـدـيـثـيـن ۴.۰ سـايـقـيـن ذـلـك: عـمـل كـيـفـيـة إـلـيـك ذـلـك، وـمـع

```
try {
    Class.forName("com.mysql.cj.jdbc.Driver"); // MySQL
    // PostgreSQL: Class.forName("org.postgresql.Driver");
} catch (ClassNotFoundException e) {
    e.printStackTrace();
}
```

اتصال تأسيس 3: الخطوة

تنسيق مرور. وكلمة مستخدم، اسم اتصال، عنوان تقطيّم طريق عن البيانات قاعدة إلى لتوصيّل DriverManager فئة استخد البيانات. قاعدة على يعتمد العنوان

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class JdbcExample {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/mydatabase"; // MySQL
        String username = "your_username";
        String password = "your_password";

        try (Connection connection = DriverManager.getConnection(url, username, password)) {
            System.out.println("!");
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

عنوانين أمثلة:

- MySQL: jdbc:mysql://localhost:3306/mydatabase?useSSL=false
- PostgreSQL: jdbc:postgresql://localhost:5432/mydatabase
- Oracle: jdbc:oracle:thin:@localhost:1521:xe

استعلامات وإجراء بيانات إنشاء 4: الخطوة

لل استعلامات ال مفضل Statement، PreparedStatement، يمكّنك الاتصال، بعد استخدامه تنفيذ المخزنة. إجراءات CallableStatement أو المعلمة

بسیط: استعلام مع مثال

```
import java.sql.*;

public class JdbcExample {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/mydatabase";
```

```

String username = "your_username";
String password = "your_password";

try (Connection connection = DriverManager.getConnection(url, username, password)) {
    Statement statement = connection.createStatement();
    ResultSet resultSet = statement.executeQuery("SELECT * FROM users");

    while (resultSet.next()) {
        System.out.println("ID: " + resultSet.getInt("id") + ", Name: " + resultSet.getString("name"));
    }
} catch (SQLException e) {
    e.printStackTrace();
}
}
}

```

المسنونات أدب الأمان لـ SQL مع مثال

```

try (Connection connection = DriverManager.getConnection(url, username, password)) {
    String sql = "SELECT * FROM users WHERE id = ?";
    PreparedStatement preparedStatement = connection.prepareStatement(sql);
    preparedStatement.setInt(1, 5); //
    ResultSet resultSet = preparedStatement.executeQuery();

    while (resultSet.next()) {
        System.out.println("ID: " + resultSet.getInt("id") + ", Name: " + resultSet.getString("name"));
    }
} catch (SQLException e) {
    e.printStackTrace();
}
}

```

الموارد إغلاق: 5 الخطوة

موضح هو كمما في استخدام الموارد. لتحري Connection و Statement، ResultSet لائينات دائمة أغلاق لك. تلقائيًّا ذلك يدير أعلى اهتمام

يعمل كامل المثال

جدول users مع id عموديًّا name عموديًّا:

```

import java.sql.*;

public class JdbcExample {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/mydatabase?useSSL=false";
        String username = "root";
        String password = "your_password";

        try (Connection connection = DriverManager.getConnection(url, username, password)) {
            String sql = "SELECT * FROM users WHERE id = ?";
            PreparedStatement preparedStatement = connection.prepareStatement(sql);
            preparedStatement.setInt(1, 1);
            ResultSet resultSet = preparedStatement.executeQuery();

            while (resultSet.next()) {
                System.out.println("ID: " + resultSet.getInt("id") + ", Name: " + resultSet.getString("name"));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}

```

رئيسيّة نصائح

- ١. **الاستثناءات:** `SQLException` هي استثناء تُرجع إلى أي خطأ وقع في عملية SQL.
- ٢. **الإعدادات:** `PreparedStatement` هي إعدادات مخصصة لتنفيذ إسقاطات SQL.
- ٣. **النافذة:** `ResultSet` هي نافذة تتيح الوصول إلى البيانات المنسوبة.
- ٤. **ال.DriverManager:** `DriverManager` هو دليل ي-handle الdrivers.

تعقيـدـاً! أكـثـرـ اسـتـخـدامـ حـالـاتـ أوـ مـعـيـنـةـ بـيـانـاتـ قـاعـدـةـ مـعـ مـسـاعـدـةـ إـلـىـ تـحـتـاجـ كـنـتـ إـذـاـ أـعـلـمـنـيـ