資料庫的操作搭配 JTable

簡介

資料庫的操作搭配 JTable 可以說是絕配。

範例

*跟 GUI 畫面整合一起提供瀏覽與搜尋之功能

前面的課程時作以下這個程式，請改成使用 JTable：

請改成使用 JTable(簡單版)

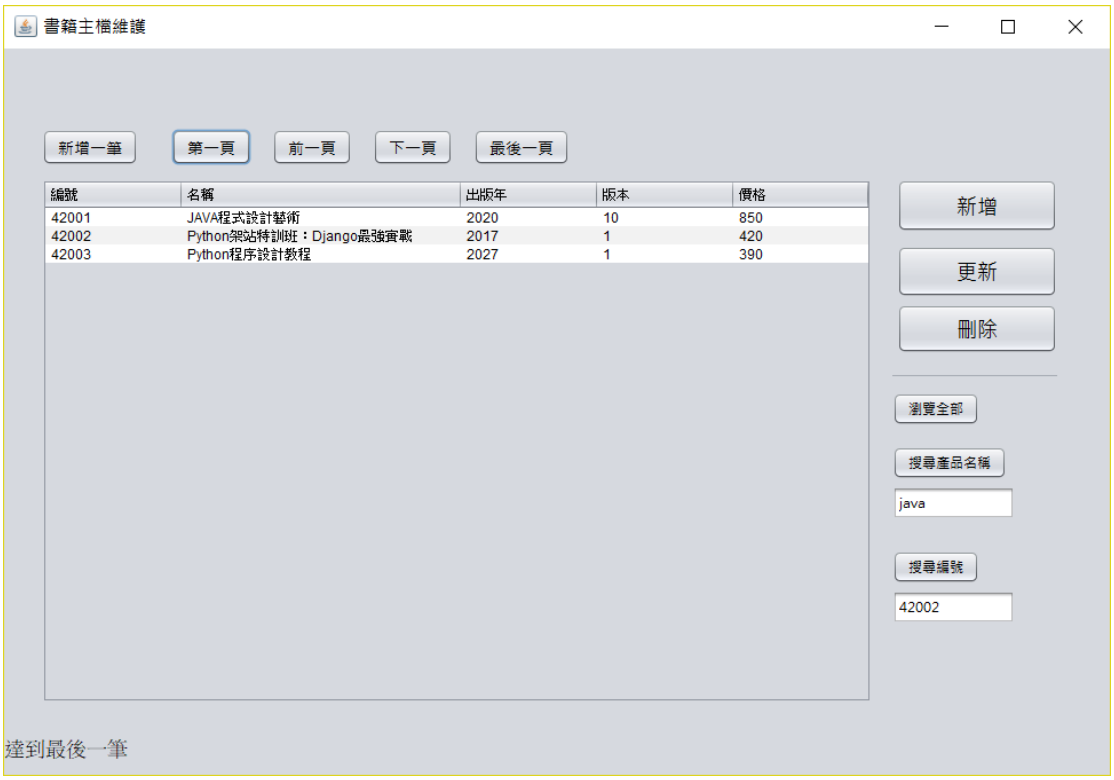
請自行上網查找解答(Youtube 也有教學)。

http://1bestcsharp.blogspot.tw/2016/01/java-and-mysql-insert-update-delete-display.html

參考程式碼:

	省略!

請改成使用 JTable 有分頁的功能(較複雜)



參考程式碼:

```
public class BookGuiTable extends javax.swing.JFrame {

    Connection conn = null;
    Statement state = null;
    ResultSet result = null;

    private int rows_per_page = 3;

    DefaultTableModel tableModel;
    private TableRowSorter rowSorter;
```

```
    public BookGuiTable() {
        initComponents();

        jTableProduct.setAutoscrolls(true);

        //jTableProduct.getColumnModel().getColumn(0).setPreferredWidth(20);
        jTableProduct.getColumnModel().getColumn(1).setPreferredWidth(200);
        //jTableProduct.getColumnModel().getColumn(2).setPreferredWidth(20);

        tableModel = (DefaultTableModel) jTableProduct.getModel();
        rowSorter = new TableRowSorter(tableModel);
        jTableProduct.setRowSorter(rowSorter);

        try {
            conn = get_conn();
            state = conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATABLE);
            result = state.executeQuery("select * from book");
            //pageinfo();
            next();
        } catch (SQLException ex) {
            System.out.println("資料庫操作出問題:" + ex.toString());
        }

    }

    public Connection get_conn() throws SQLException {
        String DB_URL = "jdbc:mariadb://localhost:3306/bookdb?user=root&password=tomtom";
        Connection conn = DriverManager.getConnection(DB_URL);
        System.out.printf("連結成功:%s\n", conn);
        return conn;
    }

    public void next() throws SQLException {
        String data[] = new String[5];

        if (result.isLast() || result.isAfterLast()) {
            System.out.println("The Last!");
```

```
            return;
        } else {
            while (tableModel.getRowCount() != 0) {
                tableModel.removeRow(0);
            }
        }

        for (int i = 1; i <= rows_per_page; i++) {

            result.relative(1);
            if (result.isAfterLast()) {
                System.out.println("The Last!");
                message.setText("達到最後一筆");
                System.out.println(result.getRow());
                break;
            }
            data[0] = result.getString(1);
            data[1] = result.getString(2);
            data[2] = result.getString(3);
            data[3] = result.getString(4);
            data[4] = result.getString(5);
            tableModel.addRow(data);
            System.out.println(result.getRow());
        }

    }

    public void previous() throws SQLException {

        String data[] = new String[5];

        if (result.isFirst() || result.isBeforeFirst()) {
            System.out.println("The First!");
            return;
        } else {
            while (tableModel.getRowCount() != 0) {
                tableModel.removeRow(0);
            }
```

```
        }

        result.relative(-rows_per_page * 2);

        if (result.isFirst() || result.isBeforeFirst()) {
            System.out.println("The First!");
        }

        for (int i = 1; i <= rows_per_page; i++) {
            result.relative(1);
            data[0] = result.getString(1);
            data[1] = result.getString(2);
            data[2] = result.getString(3);
            data[3] = result.getString(4);
            data[4] = result.getString(5);
            tableModel.addRow(data);
            //tableModel.insertRow(0, data);

            System.out.println(result.getRow());
        }
    }


    private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {

         if (jTableProduct.getSelectedRow() == -1) {
            JOptionPane.showMessageDialog(rootPane, "請先新增一筆紀錄!");
        } else {
            insert();
        }


    }

    private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
        if (jTableProduct.getSelectedRow() == -1) {
            JOptionPane.showMessageDialog(rootPane, "請選擇某一筆!");
        } else {
```

```
            update();
            JOptionPane.showMessageDialog(rootPane, "更新完成!");
        }
    }

    public void insert() {

        String id = tableModel.getValueAt(jTableProduct.getSelectedRow(), 0).toString();
        String title = tableModel.getValueAt(jTableProduct.getSelectedRow(), 1).toString();
        String pub_year = tableModel.getValueAt(jTableProduct.getSelectedRow(), 2).toString();
        String edition = tableModel.getValueAt(jTableProduct.getSelectedRow(), 3).toString();
        String price = tableModel.getValueAt(jTableProduct.getSelectedRow(), 4).toString();

        String sql = String.format("Insert into book(book_id,title,pub_year, edition, price) "
                + "values ('%s','%s','%s','%s','%s')", id, title, pub_year, edition, price);
        try {
            state.execute(sql);
            JOptionPane.showMessageDialog(rootPane, "新增完成!");
            result = state.executeQuery("select * from book");
            next();
        } catch (SQLException ex) {
            System.out.println("資料庫操作出問題:" + ex.toString());
        }
    }

    public void delete() {
        String id = tableModel.getValueAt(jTableProduct.getSelectedRow(), 0).toString();
        String sql = String.format("delete from book where book_id='%s'", id);

        try {
            state.execute(sql);
            result = state.executeQuery("select * from book");
            next();
            JOptionPane.showMessageDialog(rootPane, "刪除完成!");
        } catch (SQLException ex) {
            System.out.println("資料庫操作出問題:" + ex.toString());
        }
```

```
    }

    public void update() {
        String id = tableModel.getValueAt(jTableProduct.getSelectedRow(), 0).toString();
        String title = tableModel.getValueAt(jTableProduct.getSelectedRow(), 1).toString();
        String pub_year = tableModel.getValueAt(jTableProduct.getSelectedRow(), 2).toString();
        String edition = tableModel.getValueAt(jTableProduct.getSelectedRow(), 3).toString();
        String price = tableModel.getValueAt(jTableProduct.getSelectedRow(), 4).toString();

        String sql = String.format("update    book SET title='%s', pub_year ='%s', edition='%s', price='%s' where
book_id='%s'",
                title, pub_year, edition, price, id);
        try {
            state.executeUpdate(sql);
            System.out.println(result.getRow());
            JOptionPane.showMessageDialog(rootPane, "更新完成!");

            int current = result.getRow();
            int offset = jTableProduct.getSelectedRow();
            result.relative(offset - rows_per_page + 1);
            result.refreshRow();

            result.absolute(current - rows_per_page);
            next();

        } catch (SQLException e) {
            System.out.println("資料庫操作出問題:" + e.toString());
        }
    }

    private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
        int result = JOptionPane.showConfirmDialog(rootPane, "你要刪除這一筆嗎?", "確認",
JOptionPane.YES_NO_OPTION, JOptionPane.WARNING_MESSAGE);
        if (result == JOptionPane.YES_NO_OPTION) {
            delete();
        }
    }
```

```
private void btnNextActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        next();
    } catch (SQLException ex) {
        System.out.println("下一頁面錯誤!");
    }

}

private void btnPreviousActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        previous();
    } catch (SQLException ex) {
        System.out.println("下一頁面錯誤!");
    }
}

private void btnLastActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        result.last();
        result.relative(-rows_per_page);
        next();
        System.out.println("The Last!");
    } catch (SQLException ex) {
        System.out.println("錯誤!");
    }
}

private void btnFirstActionPerformed(java.awt.event.ActionEvent evt) {

    try {
        result.beforeFirst();
        next();
        //result.afterLast();
        System.out.println("The First!");
    } catch (SQLException ex) {
        System.out.println("錯誤!");
    }
```

```
}

private void btnNewRecActionPerformed(java.awt.event.ActionEvent evt) {

    Object arow[] = {"123", "測試書籍", "2015", "1", "300"};
    tableModel.addRow(arow);

}

private void btnFindIdActionPerformed(java.awt.event.ActionEvent evt) {

    String sql = String.format("select * from book where book_id=%s",
            tfield_id.getText());
    try {
        result = state.executeQuery(sql);
        if (result.next() == false) {
            System.out.println("無紀錄");
            JOptionPane.showMessageDialog(rootPane, "查無資料!");
        } else {
            result.beforeFirst();
            next();
        }
    } catch (SQLException ex) {
        System.out.println("檢索錯誤!");
    }
}

private void btnFindAllActionPerformed(java.awt.event.ActionEvent evt) {

    try {
        result = state.executeQuery("select * from book");
        next();
    } catch (SQLException ex) {
        System.out.println("檢索錯誤!");
    }
}
```

```
        private void btnFindTitleActionPerformed(java.awt.event.ActionEvent evt) {
            String text = tfield_title.getText();
            if (text.length() != 0) {
                try {
                    String sql = String.format("select * from book where title like '%s%%'", text);
                    result = state.executeQuery(sql);
                    next();
                } catch (SQLException ex) {
                    System.out.println("檢索錯誤!");
                }
            }
        }
}
```