Assignment 2 :-

**1. Connect to the database & create a Book table**

**2. Perform the operations of insertion, updation, deletion & query on Book table.**

**3. Write a PL/SQL code to perform the changes in price of books. For books with price above Rs. 500, increase the price by 10%.**

**For books with price below Rs. 500, increase the price by 5%.**

**# Book.java :-**

package Assignment2;

public class Book {

int bookId;

String bookName;

String authorNames;

String publication;

String dateOfPublication;

float priceOfBook;

int totalQuantityToOrder;

Book(){}

Book(int bookId,String bookName,String authorNames,String publication,String dateOfPublication,float priceOfBook,int totalQuantityToOrder)

{

this.bookId = bookId;

this.bookName = bookName;

this.authorNames = authorNames;

this.publication = publication;

this.dateOfPublication = dateOfPublication;

this.priceOfBook = priceOfBook;

this.totalQuantityToOrder = totalQuantityToOrder;

}

public int getBookId() {

return bookId;

}

public void setBookId(int bookId) {

this.bookId = bookId;

}

public String getBookName() {

return bookName;

}

public void setBookName(String bookName) {

this.bookName = bookName;

}

public String getAuthorNames() {

return authorNames;

}

public void setAuthorNames(String authorNames) {

this.authorNames = authorNames;

}

public String getPublication() {

return publication;

}

public void setPublication(String publication) {

this.publication = publication;

}

public String getDateOfPublication() {

return dateOfPublication;

}

public void setDateOfPublication(String dateOfPublication) {

this.dateOfPublication = dateOfPublication;

}

public float getPriceOfBook() {

return priceOfBook;

}

public void setPriceOfBook(float priceOfBook) {

this.priceOfBook = priceOfBook;

}

public int getTotalQuantityToOrder() {

return totalQuantityToOrder;

}

public void setTotalQuantityToOrder(int totalQuantityToOrder) {

this.totalQuantityToOrder = totalQuantityToOrder;

}

}

**# BookList.java :-**

package Assignment2;

import java.sql.\*;

import java.util.\*;

public class BookList

{

Connection con;

Statement st;

ResultSet rs;

BookList()

{}

List<Book> readBookList()

{

List<Book> books = new ArrayList<>();

try

{

Class.forName("com.mysql.jdbc.Driver");

con = DriverManager.getConnection("jdbc:mysql://localhost:3306/books?characterEncoding=latin1", "root", "khushi");

st = con.createStatement();

rs = st.executeQuery("select \* from book");

while(rs.next())

{

Book book = new Book();

book.setBookId(rs.getInt(1));

book.setBookName(rs.getString(2));

book.setAuthorNames(rs.getString(3));

book.setPublication(rs.getString(4));

book.setDateOfPublication(rs.getString(5));

book.setPriceOfBook(rs.getFloat(6));

book.setTotalQuantityToOrder(rs.getInt(7));

books.add(book);

}

rs.close();

st.close();

con.close();

}

catch (Exception exp)

{

exp.printStackTrace();

}

return books;

}

}

**# DisplayFrame.java :-**

package Assignment2;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.List;

public class DisplayFrame extends JFrame

{

List<Book> books;

JPanel mainPanel;

JScrollPane scrollPane;

JLabel headingLabel;

JButton addBookButton;

JButton incPriceButton;

public DisplayFrame(List<Book> books)

{

this.books = books;

this.mainPanel = new JPanel();

this.headingLabel = new JLabel("Books Display", JLabel.***CENTER***);

this.addBookButton = new JButton("Add New Book");

this.incPriceButton = new JButton("Increase Price of Book");

scrollPane = new JScrollPane(mainPanel);

createFrame();

}

void createFrame()

{

createHeaderPanel();

createMainPanel();

createBookPanels();

setSize(800, 1000);

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setLocationRelativeTo(null);

add(scrollPane);

setVisible(true);

}

void createHeaderPanel()

{

JPanel headerPanel = new JPanel();

headerPanel.setLayout(new BorderLayout());

headerPanel.setBackground(new Color(245, 245, 250));

headingLabel.setFont(new Font("Arial", Font.***BOLD***, 28));

headingLabel.setForeground(new Color(0, 0, 0));

headerPanel.add(headingLabel, BorderLayout.***CENTER***);

addBookButton = new JButton("+");

addBookButton.setFont(new Font("Arial", Font.***BOLD***, 28));

addBookButton.setPreferredSize(new Dimension(50, 50));

addBookButton.addActionListener(new ActionListener()

{

*@Override*

public void actionPerformed(ActionEvent e)

{

InsertionFrame insertionFrame = new InsertionFrame();

}

});

incPriceButton = new JButton("↑");

incPriceButton.setFont(new Font("Arial", Font.***BOLD***, 30));

incPriceButton.setPreferredSize(new Dimension(50, 50));

incPriceButton.addActionListener(new ActionListener()

{

*@Override*

public void actionPerformed(ActionEvent e)

{

IncreasePrice increasePrice = new IncreasePrice();

JOptionPane.*showMessageDialog*(null, "Book prices have been updated!", "Price Update", JOptionPane.***INFORMATION\_MESSAGE***);

refreshFrame();

}

});

JPanel buttonPanelRight = new JPanel();

buttonPanelRight.setBackground(new Color(245, 245, 250));

buttonPanelRight.add(addBookButton);

JPanel buttonPanelLeft = new JPanel();

buttonPanelLeft.setBackground(new Color(245, 245, 250));

buttonPanelLeft.add(incPriceButton);

headerPanel.add(buttonPanelRight, BorderLayout.***EAST***);

headerPanel.add(buttonPanelLeft, BorderLayout.***WEST***);

this.add(headerPanel, BorderLayout.***NORTH***);

}

void createMainPanel()

{

mainPanel.add(Box.*createVerticalStrut*(20));

mainPanel.setLayout(new BoxLayout(mainPanel, BoxLayout.***Y\_AXIS***));

mainPanel.setBackground(new Color(245, 245, 250));

mainPanel.setBorder(BorderFactory.*createEmptyBorder*(15, 15, 15, 15));

}

void refreshFrame()

{

BookList booklist = new BookList();

books = booklist.readBookList();

mainPanel.removeAll();

createBookPanels();

mainPanel.revalidate();

mainPanel.repaint();

}

void createBookPanels()

{

for (Book book : books)

{

BookPanel bookPanel = new BookPanel(book);

mainPanel.add(bookPanel);

mainPanel.add(Box.*createVerticalStrut*(15));

}

}

}

**# InsertFrame.java :-**

package Assignment2;

import java.awt.\*;

import java.awt.event.\*;

import java.io.\*;

import javax.swing.\*;

public class InsertionFrame extends JFrame

{

JLabel headerLabel;

JPanel panel;

String labelText[] = {"Book ID:","Book Name:","Author Names:","Publication:","Date of Publication:","Price of Book:","Total Quantity to Order:"};

JLabel label[];

JTextField textField[];

JButton submitButton;

JPanel footerPanel;

public InsertionFrame()

{

this.headerLabel = new JLabel("Insert New Book Details", JLabel.***CENTER***);

this.panel = new JPanel();

this.label = new JLabel[labelText.length];

this.textField = new JTextField[labelText.length];

this.submitButton = new JButton("Insert Book");

this.footerPanel = new JPanel();

createFrame();

}

void createFrame()

{

this.setBackground(new Color(245, 239, 255));

this.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

this.setSize(700, 600);

this.setLayout(new BorderLayout(10, 10));

createComponents();

this.add(headerLabel, BorderLayout.***NORTH***);

addingComponentsToPanel();

this.add(panel, BorderLayout.***CENTER***);

this.add(footerPanel, BorderLayout.***SOUTH***);

this.setVisible(true);

}

void createComponents()

{

createHeader();

createPanel();

createLabel();

createSubmitButton();

createFooter();

}

void createHeader()

{

headerLabel.setFont(new Font("Arial", Font.***BOLD***, 24));

headerLabel.setBorder(BorderFactory.*createEmptyBorder*(10, 0, 10, 0));

}

void createPanel()

{

panel.setLayout(new GridLayout(7, 2, 10, 10));

panel.setBorder(BorderFactory.*createEmptyBorder*(20, 40, 20, 40));

}

void createLabel()

{

for(int i=0; i<label.length; i++)

{

label[i] = new JLabel(labelText[i]);

textField[i] = new JTextField();

}

}

void createSubmitButton()

{

submitButton.setForeground(Color.***WHITE***);

submitButton.setPreferredSize(new Dimension(300, 40));

submitButton.setBackground(new Color(146, 145, 194));

submitButton.setFont(new Font("Arial", Font.***BOLD***, 16));

submitButton.setFocusPainted(false);

submitButton.addActionListener(new ActionListener() {

*@Override*

public void actionPerformed(ActionEvent e) {

Book b = new Book(Integer.*parseInt*(textField[0].getText()), textField[1].getText(), textField[2].getText(), textField[3].getText(), textField[4].getText(), Float.*parseFloat*(textField[5].getText()), Integer.*parseInt*(textField[6].getText()));

BookInsertion bookInsertion = new BookInsertion(b);

bookInsertion.insertBook();

showMessageDialog(b);

}});

}

void showMessageDialog(Book b)

{

JOptionPane.*showMessageDialog*(this, "Book Inserted:\n" +

"Book ID: " + b.getBookId() + "\n" +

"Book Name: " + b.getBookName() + "\n" +

"Author Names: " + b.getAuthorNames() + "\n" +

"Publication: " + b.getPublication() + "\n" +

"Date of Publication: " + b.getDateOfPublication() + "\n" +

"Price of Book: " + b.getPriceOfBook() + "\n" +

"Total Quantity to Order: " + b.getTotalQuantityToOrder());

}

void createFooter()

{

footerPanel.setBorder(BorderFactory.*createEmptyBorder*(20, 0, 20, 0));

footerPanel.add(submitButton);

}

void addingComponentsToPanel()

{

for(int i=0; i<label.length; i++)

{

panel.add(label[i]);

panel.add(textField[i]);

}

}

}

**# UpdateFrame.java :-**

package Assignment2;

import java.awt.BorderLayout;

import java.awt.Color;

import java.awt.Dimension;

import java.awt.Font;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileWriter;

import java.util.List;

import javax.swing.BorderFactory;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class UpdateFrame extends JFrame

{

Book book;

List<Book> books;

BookList bookList;

File file;

JLabel headerLabel;

JPanel panel;

String labelText[] = {"Book ID:","Book Name:","Author Names:","Publication:","Date of Publication:","Price of Book:","Total Quantity to Order:"};

JLabel label[];

JTextField textField[];

JButton submitButton;

JPanel footerPanel;

public UpdateFrame(Book book)

{

this.book = book;

this.bookList = new BookList();

this.books = bookList.readBookList();

this.file = new File("book\_list.dat");

this.headerLabel = new JLabel("Book Details", JLabel.CENTER);

this.panel = new JPanel();

this.label = new JLabel[labelText.length];

this.textField = new JTextField[labelText.length];

this.submitButton = new JButton("Update Book");

this.footerPanel = new JPanel();

createFrame();

}

void createFrame()

{

this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

this.setSize(700, 600);

this.setLayout(new BorderLayout(10, 10));

createComponents();

this.add(headerLabel, BorderLayout.NORTH);

addingComponentsToPanel();

this.add(panel, BorderLayout.CENTER);

this.add(footerPanel, BorderLayout.SOUTH);

this.setVisible(true);

}

void createComponents()

{

createHeader();

createPanel();

createLabel();

createSubmitButton();

createFooter();

}

void createHeader()

{

headerLabel.setFont(new Font("Arial", Font.BOLD, 24));

headerLabel.setBorder(BorderFactory.createEmptyBorder(10, 0, 10, 0));

}

void createPanel()

{

panel.setLayout(new GridLayout(7, 2, 10, 10));

panel.setBorder(BorderFactory.createEmptyBorder(20, 40, 20, 40));

}

void createLabel()

{

for(int i=0; i<label.length; i++)

{

label[i] = new JLabel(labelText[i]);

textField[i] = new JTextField();

}

createTextField();

}

void createTextField()

{

textField[0].setEditable(false);

textField[0].setText(""+book.getBookId());

textField[1].setText(""+book.getBookName());

textField[2].setText(""+book.getAuthorNames());

textField[3].setText(""+book.getPublication());

textField[4].setText(""+book.getDateOfPublication());

textField[5].setText(""+book.getPriceOfBook());

textField[6].setText(""+book.getTotalQuantityToOrder());

}

void createSubmitButton()

{

submitButton.setPreferredSize(new Dimension(300, 40));

submitButton.setBackground(new Color(146, 145, 194));

submitButton.setForeground(Color.WHITE);

submitButton.setFont(new Font("Arial", Font.BOLD, 16));

submitButton.setFocusPainted(false);

submitButton.addActionListener(new ActionListener()

{

@Override

public void actionPerformed(ActionEvent e)

{

Book b = new Book(Integer.parseInt(textField[0].getText()), textField[1].getText(), textField[2].getText(), textField[3].getText(), textField[4].getText(), Float.parseFloat(textField[5].getText()), Integer.parseInt(textField[6].getText()));

BookUpdatation bookUpdatation = new BookUpdatation(b);

bookUpdatation.updateBook();

showMessageDialog();

}

});

}

void showMessageDialog()

{

JOptionPane.showMessageDialog(this, "Book Updated Successfully...!");

}

void createFooter()

{

footerPanel.setBorder(BorderFactory.createEmptyBorder(20, 0, 20, 0));

footerPanel.add(submitButton);

}

void addingComponentsToPanel()

{

for(int i=0; i<label.length; i++)

{

panel.add(label[i]);

panel.add(textField[i]);

}

}

}

**# BookPanel.java :-**

package Assignment2;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.\*;

class BookPanel extends JPanel

{

Book book;

JPanel detailsPanel;

JPanel buttonPanel;

JButton updateButton;

JButton deleteButton;

JLabel labels[];

BookPanel(Book book)

{

this.book = book;

setLayout(new BorderLayout(10, 5));

detailsPanel = new JPanel(new GridLayout(4, 2, 25, 15));

buttonPanel = new JPanel(new FlowLayout(FlowLayout.***CENTER***,20,10));

labels = createLabels();

deleteButton = new JButton("Delete");

updateButton = new JButton("Update");

createBookPanel();

}

void createBookPanel()

{

setBackground(new Color(245, 239, 255));

setBorder(BorderFactory.*createCompoundBorder*(

BorderFactory.*createLineBorder*(new Color(173, 173, 209), 3),

BorderFactory.*createEmptyBorder*(20, 20, 20, 20)));

detailsPanel.setBackground(new Color(245, 239, 255));

buttonPanel.setBackground(new Color(245, 239, 255));

for (JLabel label : labels)

{

label.setFont(new Font("Arial", Font.***BOLD***, 14));

detailsPanel.add(label);

}

updateDeleteButton();

add(detailsPanel, BorderLayout.***CENTER***);

add(buttonPanel, BorderLayout.***SOUTH***);

}

void updateDeleteButton()

{

updateButton.setBackground(new Color(146, 145, 194));

deleteButton.setBackground(new Color(146, 145, 194));

updateButton.setFont(new Font("Arial", Font.***BOLD***, 14));

deleteButton.setFont(new Font("Arial", Font.***BOLD***, 14));

updateButton.setForeground(Color.***WHITE***);

deleteButton.setForeground(Color.***WHITE***);

updateButton.setPreferredSize(new Dimension(300, 40));

deleteButton.setPreferredSize(new Dimension(300, 40));

updateButton.addActionListener(new ActionListener()

{

*@Override*

public void actionPerformed(ActionEvent e)

{

JOptionPane.*showMessageDialog*(null, "Update button clicked for Book ID: " + book.getBookId());

UpdateFrame updateFrame = new UpdateFrame(book);

}

});

deleteButton.addActionListener(new ActionListener()

{

*@Override*

public void actionPerformed(ActionEvent e)

{

BookDeletion bookDeletion = new BookDeletion(book);

if(bookDeletion.showDeleteConfirmation()==JOptionPane.***YES\_OPTION***)

{

bookDeletion.deleteBook();

Container parent = getParent();

if (parent != null)

{

parent.remove(BookPanel.this);

parent.revalidate();

parent.repaint();

}

}

}

});

buttonPanel.add(updateButton);

buttonPanel.add(deleteButton);

}

JLabel[] createLabels()

{

return new JLabel[]

{

new JLabel("Book ID: " + book.getBookId()),

new JLabel("Book Name: " + book.getBookName()),

new JLabel("Author: " + book.getAuthorNames()),

new JLabel("Publication: " + book.getPublication()),

new JLabel("Date: " + book.getDateOfPublication()),

new JLabel("Price: " + book.getPriceOfBook()),

new JLabel("Quantity: " + book.getTotalQuantityToOrder()),

new JLabel("Total Cost: " + book.getTotalQuantityToOrder()\*book.getPriceOfBook()),

};

}

public void refreshFrame()

{

this.revalidate();

this.repaint();

}

Book getBook()

{

return book;

}

}

**# BookDisplay.java :-**

package Assignment2;

import java.util.List;

public class BookDisplay

{

public static void main(String[] args)

{

BookList booklist = new BookList();

List<Book> books = booklist.readBookList();

DisplayFrame frame = new DisplayFrame(books);

}

}

**# BookInsertion.java :-**

package Assignment2;

import java.sql.\*;

public class BookInsertion

{

Connection con;

Book book;

PreparedStatement prs;

CallableStatement cst;

public BookInsertion(Book book)

{

this.book = book;

}

void insertBook()

{

try

{

Class.*forName*("com.mysql.jdbc.Driver");

con = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/books?characterEncoding=latin1", "root", "khushi");

prs = con.prepareStatement("insert into book (bookId, bookName, authorNames, publication, dateOfPublication, priceOfBook, totalQuantityToOrder) values(?,?,?,?,?,?,?)");

prs.setInt(1, book.getBookId());

prs.setString(2, book.getBookName());

prs.setString(3, book.getAuthorNames());

prs.setString(4, book.getPublication());

prs.setString(5, book.getDateOfPublication());

prs.setDouble(6, book.getPriceOfBook());

prs.setInt(7, book.getTotalQuantityToOrder());

prs.execute();

prs.close();

cst = con.prepareCall("{ call set\_total\_cost(?) }");

cst.setInt(1, book.getBookId());

cst.execute();

cst.close();

con.close();

}

catch (Exception e)

{

e.printStackTrace();

}

}

}

**# BookDeletion.java :-**

package Assignment2;

import java.sql.\*;

import javax.swing.JOptionPane;

public class BookDeletion

{

Connection con;

Book book;

PreparedStatement prs;

public BookDeletion(Book book)

{

this.book = book;

}

public int showDeleteConfirmation()

{

return JOptionPane.showConfirmDialog(

null,

"Are you sure you want to remove this book ?",

"Confirm Deletion",

JOptionPane.YES\_NO\_OPTION,

JOptionPane.WARNING\_MESSAGE

);

}

void deleteBook()

{

try

{

Class.forName("com.mysql.jdbc.Driver");

con = DriverManager.getConnection("jdbc:mysql://localhost:3306/books?characterEncoding=latin1", "root", "khushi");

prs = con.prepareStatement("delete from book where bookId = ?");

prs.setInt(1, book.getBookId());

prs.execute();

prs.close();

con.close();

}

catch (Exception e)

{

e.printStackTrace();

System.out.println("Error writing to file!");

}

}

}

**# BookUpdation.java :-**

package Assignment2;

import java.sql.\*;

public class BookUpdatation

{

Connection con;

Book book;

PreparedStatement prs;

CallableStatement cst;

public BookUpdatation(Book book)

{

this.book = book;

}

void updateBook()

{

try

{

Class.*forName*("com.mysql.jdbc.Driver");

con = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/books?characterEncoding=latin1", "root", "khushi");

prs = con.prepareStatement("update book set bookName = ?, authorNames = ?, publication = ?, dateOfPublication = ?, priceOfBook = ?, totalQuantityToOrder = ? where bookId = ?");

prs.setString(1, book.getBookName());

prs.setString(2, book.getAuthorNames());

prs.setString(3, book.getPublication());

prs.setString(4, book.getDateOfPublication());

prs.setDouble(5, book.getPriceOfBook());

prs.setInt(6, book.getTotalQuantityToOrder());

prs.setInt(7, book.getBookId());

prs.execute();

prs.close();

cst = con.prepareCall("{ call set\_total\_cost(?) }");

cst.setInt(1, book.getBookId());

cst.execute();

cst.close();

con.close();

}

catch (Exception e)

{

e.printStackTrace();

}

}

}

**#IncreasePrice.java :-**

package Assignment2;

import java.sql.CallableStatement;

import java.sql.Connection;

import java.sql.DriverManager;

import javax.swing.JOptionPane;

public class IncreasePrice

{

Connection con;

CallableStatement cst;

public IncreasePrice()

{

increasePriceOfBooks();

}

void increasePriceOfBooks()

{

try

{

Class.*forName*("com.mysql.jdbc.Driver");

con = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/books?characterEncoding=latin1", "root", "khushi");

cst = con.prepareCall("{ call update\_book\_prices() }");

cst.execute();

cst.close();

con.close();

}

catch (Exception e)

{

e.printStackTrace();

}

}

}