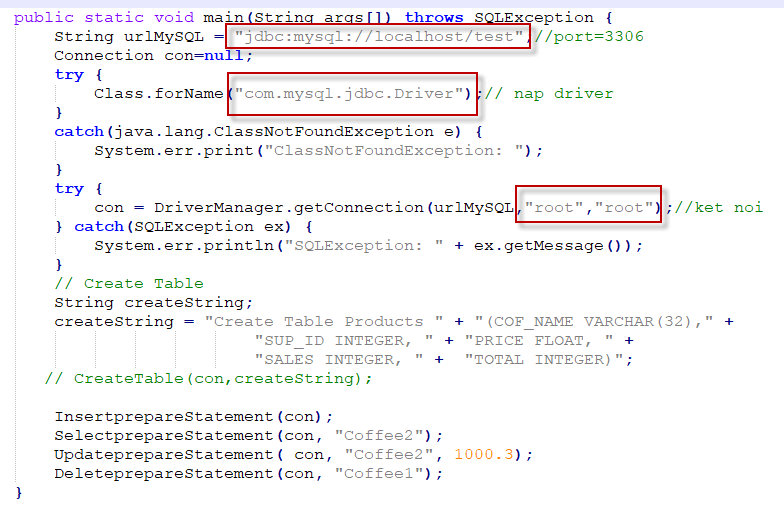
## JDBC

## 1. Cấu hình

### Cấu hình driver



### Kết nối CSDL



## 2. Thực hiện các thao tác

### 2.1. Tạo bảng

public static void CreateTable(Connection conn, String createTable) throws SQLException{

Statement stmt;

stmt = conn.createStatement();// buoc 3

stmt.executeUpdate(createTable);

stmt.close();

}

### 2.2. Chèn dữ liệu

public static void InsertprepareStatement(Connection conn){

String insertQuery = "INSERT INTO Products VALUES(?,?,?,?,?);";

if (conn != null){

try{

PreparedStatement prest = conn.prepareStatement(insertQuery);

prest.setString(1, "Coffee1");

prest.setInt(2, 2);

prest.setDouble(3, 3.1);

prest.setInt(4, 2);

prest.setInt(5, 2);

prest.executeUpdate();

prest.setString(1, "Coffee2");

prest.setInt(2, 2);

prest.setDouble(3, 3.1);

prest.setInt(4, 2);

prest.setInt(5, 2);

prest.executeUpdate();

prest.close();

}catch(SQLException e){

e.printStackTrace();

}

}

}

### 2.3. Truy vấn

public static void SelectprepareStatement(Connection conn, String nameProduct){

String selectQuery = "Select \* from Products where COF\_NAME=?";

if (conn != null){

try{

PreparedStatement prest = conn.prepareStatement(selectQuery);

prest.setString(1, nameProduct);

ResultSet rs = prest.executeQuery();

while (rs.next()){

String name = rs.getString(1);

double val = rs.getDouble(3);

System.out.println("Name= "+name + "\t\t Price=" + val);

}

prest.close();

}catch(SQLException e){

e.printStackTrace();

}

}

}

### 2.4. Cập nhật

public static void UpdateprepareStatement(Connection conn, String nameProduct, double newPrice){

String selectQuery = "Update Products set PRICE=? where COF\_NAME=?";

if (conn != null){

try{

PreparedStatement prest = conn.prepareStatement(selectQuery);

prest.setDouble(1, newPrice);

prest.setString(2, nameProduct);

prest.executeUpdate();

prest.close();

}catch(SQLException e){

e.printStackTrace();

}

}

}

### 2.5. Xóa

public static void DeleteprepareStatement(Connection conn, String nameProduct){

String selectQuery = "Delete from Products where COF\_NAME=?";

if (conn != null){

try{

PreparedStatement prest = conn.prepareStatement(selectQuery);

prest.setString(1, nameProduct);

prest.executeUpdate();

prest.close();

}catch(SQLException e){

e.printStackTrace();

}

}

}

## 3. Thực thi thủ tục

### 3.1. Tạo DB, Procedures

Create Database SampleDB // My SQL

go

use SampleDB;

go

CREATE TABLE authors (

author\_id int(11) NOT NULL AUTO\_INCREMENT,

name varchar(45) NOT NULL,

email varchar(45) NOT NULL,

PRIMARY KEY (author\_id)

);

CREATE TABLE books (

book\_id int(11) NOT NULL AUTO\_INCREMENT,

title varchar(200) NOT NULL,

description varchar(200) NOT NULL,

published date NULL,

author\_id int(11) NULL,

price float,

rating int(11),

PRIMARY KEY (book\_id)

);

CREATE PROCEDURE Create\_author (IN name VARCHAR(45), email VARCHAR(45))

BEGIN

DECLARE newAuthorID INT;

INSERT INTO authors (name, email) VALUES (name, email);

SET newAuthorID = (SELECT author\_id FROM authors a WHERE a.name = name);

INSERT INTO books (title, description, published, author\_id, price, rating)

VALUES (CONCAT('Life Story of ', name),

CONCAT('Personal Stories of ', name),

date('2016-12-30'), newAuthorID, 10.00, 0);

END

CREATE PROCEDURE get\_books(IN rate INT)

BEGIN

SELECT \* FROM books WHERE rating >= rate;

END

-- Cac tham so phai viet tren 1 dongs

CREATE PROCEDURE summary\_report( IN title1 VARCHAR(45), OUT totalBooks INT, OUT totalValue DOUBLE, OUT highPrice DOUBLE )

BEGIN

DECLARE maxPrice DOUBLE;

SELECT COUNT(\*) AS bookCount, SUM(price) as total

FROM books b JOIN authors a ON b.author\_id = a.author\_id

AND b.title LIKE CONCAT('%', title, '%')

INTO totalBooks, totalValue;

SELECT MAX(price) FROM books WHERE price INTO maxPrice;

IF (maxPrice > -1) THEN

SET highPrice = maxPrice;

END IF;

END

### 3.2. Thực thi thủ tục truy vấn

public class StoredProcedureCallSelect {

static {

try {

Class.forName("com.mysql.jdbc.Driver");// nap driver

} catch (java.lang.ClassNotFoundException e) {

System.err.print("ClassNotFoundException: ");

}

} ;

public static void main(String[] args) {

String dbURL = "jdbc:mysql://localhost:3306/sampleDB";

String user = "root";

String password = "root";

try (

Connection conn = DriverManager.getConnection(dbURL, user, password);

CallableStatement statement = conn.prepareCall("{call get\_books(?)}");

) {

statement.setInt(1, 0);

boolean hadResults = statement.execute();

// print headings

System.out.println("| Title | Description | Rating |");

System.out.println("================================");

while (hadResults) {

ResultSet resultSet = statement.getResultSet();

// process result set

while (resultSet.next()) {

String title = resultSet.getString("title");

String description = resultSet.getString("description");

int rating = resultSet.getInt("rating");

System.out.println(

"| " + title + " | " + description + " | " + rating + " |");

}

hadResults = statement.getMoreResults();

}

statement.close();

} catch (SQLException ex) { ex.printStackTrace(); }

}

}

### 3.3. Thực thi thủ tục chèn

public class StoredProcedureCallInsert {

static{

try {

Class.forName("com.mysql.jdbc.Driver");// nap driver

}

catch(java.lang.ClassNotFoundException e) {

System.err.print("ClassNotFoundException: ");

}

};

public static void main(String[] args) {

String dbURL = "jdbc:mysql://localhost:3306/sampleDB";

String user = "root";

String password = "root";

try (

Connection conn = DriverManager.getConnection(dbURL, user, password);

CallableStatement statement

= conn.prepareCall("{call create\_author(?, ?)}");) {

statement.setString(1, "Bill Gates");

statement.setString(2, "bill@microsoft.com");

statement.execute();

statement.close();

System.out.println("Stored procedure called successfully!");

} catch (SQLException ex) {

ex.printStackTrace();

}

}

}

### 3.4. Thực thi thủ tục có tham số output

public class StoredProcedureCallOuputParameters {

static {

try {

Class.forName("com.mysql.jdbc.Driver");// nap driver

} catch (java.lang.ClassNotFoundException e) {

System.err.print("ClassNotFoundException: ");

}

} ;

public static void main(String[] args) {

String dbURL = "jdbc:mysql://localhost:3306/sampleDB";

String user = "root";

String password = "root";

try (

Connection conn = DriverManager.getConnection(dbURL, user, password);) {

CallableStatement statement

= conn.prepareCall("{call summary\_report (?,?,?,?)}");

statement.registerOutParameter(2, Types.INTEGER);

statement.registerOutParameter(3, Types.DOUBLE);

statement.registerOutParameter(4, Types.DOUBLE);

statement.setString(1, "Java");

statement.execute();

Integer totalBook = statement.getInt(2);

Double totalValue = statement.getDouble(3);

Double highPrice = statement.getDouble("highPrice");

System.out.println("Total books: " + totalBook);

System.out.println("Total value: " + totalValue);

System.out.println("High price: " + highPrice);

statement.close();

} catch (SQLException ex) {

ex.printStackTrace();

}

}

}

### 3.5. Thực thi transaction

### 3.5.1.Tạo DB

use SampleDB;

CREATE TABLE orders (

order\_id int(11) NOT NULL AUTO\_INCREMENT,

product\_id int(11) DEFAULT NULL,

amount float DEFAULT NULL,

order\_date datetime DEFAULT NULL,

PRIMARY KEY (order\_id)

) ;

CREATE TABLE products (

product\_id int(11) NOT NULL AUTO\_INCREMENT,

product\_name varchar(45) NOT NULL,

price float DEFAULT NULL,

PRIMARY KEY (product\_id)

);

CREATE TABLE monthly\_sales (

report\_month int(11),

product\_id int(11),

total\_amount float

);

### 3.5.2. Chương trình

public class TransactDemo {

static {

try {

Class.forName("com.mysql.jdbc.Driver");// nap driver

} catch (java.lang.ClassNotFoundException e) {

System.err.print("ClassNotFoundException: ");

}

}

;

private String dbURL = "jdbc:mysql://localhost:3306/sampleDB";

private String user = "root";

private String password = "root";

private Connection conn;

public void connect() {

try {

conn = DriverManager.getConnection(dbURL, user, password);

System.out.println("Connected.");

} catch (SQLException ex) {

ex.printStackTrace();

}

}

public void disconnect() {

try {

conn.close();

System.out.println("Closed.");

} catch (SQLException ex) {

ex.printStackTrace();

}

}

public void saveOrder(int productId, Date orderDate, float amount, int reportMonth) {

PreparedStatement orderStatement = null;

PreparedStatement saleStatement = null;

try {

conn.setAutoCommit(false);

String sqlSaveOrder = "insert into orders (product\_id, order\_date, amount)";

sqlSaveOrder += " values (?, ?, ?)";

String sqlUpdateTotal = "update monthly\_sales set total\_amount = total\_amount + ?";

sqlUpdateTotal += " where product\_id = ? and report\_month = ?";

orderStatement = conn.prepareStatement(sqlSaveOrder);

saleStatement = conn.prepareStatement(sqlUpdateTotal);

orderStatement.setInt(1, productId);

orderStatement.setDate(2, orderDate);

orderStatement.setFloat(3, amount);

saleStatement.setFloat(1, amount);

saleStatement.setInt(2, productId);

saleStatement.setInt(3, reportMonth);

orderStatement.executeUpdate();

saleStatement.executeUpdate();

conn.commit();

} catch (SQLException ex) {

if (conn != null) {

try {

conn.rollback();

System.out.println("Rolled back.");

} catch (SQLException exrb) {

exrb.printStackTrace();

}

}

} finally {

try {

if (orderStatement != null) {

orderStatement.close();

}

if (saleStatement != null) {

saleStatement.close();

}

conn.setAutoCommit(true);

} catch (SQLException excs) {

excs.printStackTrace();

}

}

}

public static void main(String[] args) {

TransactDemo demo = new TransactDemo();

int productId = 1;

int reportMonth = 7;

Date date = new Date(System.currentTimeMillis());

float amount = 580;

demo.connect();

demo.saveOrder(productId, date, amount, reportMonth);

demo.disconnect();

}

}

## 4. Thực hành RowSet:

### 4.1. JdbcRowSet

### 4.2. CachedRowSet

### 4.3. JoinRowSet