Instituto Politécnico Nacional

Escuela Superior de Cómputo

Web Client and Backend Development Frameworks.

HW2 : Reporte de Lectura JDBC

Profesor: M. en C. José Asunción Enríquez Zárate

Alumno: Ivonne Mendez Cruz

ivonne.mendezw@gmail.com

7CM1

March 7, 2025

Contents

1 Resumen Ejecutivo

Este documento es una guía rápida sobre el uso de JDBC (Java Database Connectivity) para la manipulación de bases de datos desde aplicaciones Java. Presenta conceptos fundamentales de bases de datos relacionales, instrucciones para configurar y usar MySQL con JDBC, y ejemplos prácticos de consultas SQL y su integración con Java.

2 Objetivos del Capítulo

Este documento tiene como propósito que el lector aprenda:

- Conceptos fundamentales sobre bases de datos relacionales.
- Uso del lenguaje SQL para consultar y manipular datos.
- Implementación de la API JDBC en Java para conectarse a bases de datos.
- Uso de la interfaz RowSet para manejar bases de datos.
- Creación y uso de PreparedStatement para consultas eficientes y seguras.
- Implementación del procesamiento de transacciones para robustecer aplicaciones.

3 Introducción a las Bases de Datos Relacionales

Una base de datos relacional organiza datos en tablas con filas y columnas. Se mencionan algunos de los sistemas de gestión de bases de datos más utilizados:

- Microsoft SQL Server
- Oracle
- IBM DB2
- MySQL
- PostgreSQL

SQL es el lenguaje estándar internacional utilizado para realizar consultas y manipular datos en bases de datos relacionales.

4 Lenguaje SQL

Se presentan algunos ejemplos de consultas básicas y avanzadas:

4.1 Consultas Básicas (SELECT)

```sql
SELECT * FROM tableName;
SELECT column1, column2 FROM tableName;
```

4.2 Condiciones y Filtrado (WHERE)

```sql
SELECT * FROM Employees WHERE Department = 'IT';
```

4.3 Ordenación (ORDER BY)

```sql
SELECT * FROM Employees ORDER BY Salary DESC;
```

4.4 Uniones de Tablas (INNER JOIN)

```
SELECT A.Name, B.Salary FROM Employees A
INNER JOIN Salaries B ON A.ID = B.ID;
```

4.5 Manipulación de Datos

```
INSERT INTO tableName VALUES (...);
UPDATE tableName SET column = value WHERE condition;
DELETE FROM tableName WHERE condition;
```

5 Instalación y Configuración de MySQL con JDBC

El documento explica cómo instalar MySQL y configurar MySQL Connector/J para permitir la comunicación con JDBC. Se incluyen:

- Pasos para instalar MySQL.
- Creación de un usuario en MySQL.
- Ejecución de scripts SQL para crear bases de datos y tablas.

6 Uso de JDBC en Java

Ejemplo de conexión a base de datos con JDBC:

```
Connection connection = DriverManager.getConnection(
"jdbc:mysql://localhost/books", "usuario", "contrase a");
```

Ejemplo de consulta:

```
Statement statement = connection.createStatement();
ResultSet resultSet = statement.executeQuery("SELECT * FROM Employees");
while (resultSet.next()) {
System.out.println(resultSet.getString("Name"));
}
```

Cierre seguro de conexión:

```
resultSet.close();
statement.close();
connection.close();
```

7 Uso de PreparedStatement para Consultas Seguras

```
String query = "SELECT * FROM Employees WHERE Name = ?";
PreparedStatement preparedStatement = connection.prepareStatement(query);
preparedStatement.setString(1, "John");
ResultSet resultSet = preparedStatement.executeQuery();
```

8 Procesamiento de Transacciones

```
connection.setAutoCommit(false);
try {
statement.executeUpdate("INSERT INTO Orders VALUES (...)");
statement.executeUpdate("UPDATE Inventory SET Stock = Stock - 1 WHERE ID = ?");
connection.commit();
} catch (SQLException e) {
connection.rollback();
}
```

9 Conclusiones y Recomendaciones

- JDBC es una forma eficiente de conectar aplicaciones Java con bases de datos.
- PreparedStatement y transacciones mejoran la seguridad y la consistencia de datos.
- Es recomendable cerrar conexiones para evitar fugas de memoria.
- Se pueden mejorar las interfaces con JTable para visualizar datos en tablas dinámicas.
- Se hace énfasis en los tipos de errores más comunes:
  - Uso incorrecto de ResultSet.next() sin verificar si hay datos disponibles.
  - Acceso a columnas inexistentes en ResultSet.
  - No cerrar conexiones, lo que provoca fugas de memoria.
  - Errores en transacciones por mal manejo de commit y rollback.

10 Conclusión

sectionConclusión El uso de JDBC en aplicaciones Java es una herramienta poderosa para la interacción con bases de datos relacionales. Gracias a su flexibilidad y compatibilidad con múltiples sistemas de gestión de bases de datos, permite a los desarrolladores implementar soluciones robustas y escalables para la manipulación de datos.

Uno de los principales beneficios de JDBC es su capacidad de abstracción, lo que permite a las aplicaciones interactuar con diferentes bases de datos sin necesidad de modificar el código fuente de manera significativa. La implementación de PreparedStatement mejora la seguridad al prevenir inyecciones SQL, y el uso adecuado de transacciones permite mantener la integridad de los datos.

Es crucial seguir buenas prácticas, como cerrar conexiones y gestionar correctamente los errores, para evitar fugas de memoria y garantizar el correcto funcionamiento del sistema. Asimismo, la optimización de consultas y la gestión eficiente de ResultSets pueden mejorar significativamente el rendimiento de la aplicación.

En el ámbito del desarrollo empresarial, el uso de JDBC se puede complementar con frameworks más avanzados como Hibernate o JPA, que simplifican aún más la gestión de bases de datos y mejoran la productividad del desarrollo.

En conclusión, dominar JDBC y sus mejores prácticas proporciona una base sólida para el desarrollo de aplicaciones con acceso a bases de datos. A medida que los sistemas crecen en complejidad, es recomendable evaluar el uso de herramientas adicionales que faciliten la gestión de datos y mejoren la eficiencia del desarrollo. La correcta implementación de JDBC no solo garantiza el acceso eficiente a la información, sino que también contribuye a la seguridad y estabilidad de las aplicaciones.

11 Referencias

References

[1] Autor desconocido. (2025). *Acceso a Bases de Datos con JDBC*. Documento técnico sobre JDBC y bases de datos relacionales. Documento proporcionado en formato PDF.