[**오라클(Oracle) BLOB 데이터 입력 - JDBC**](http://devhome.tistory.com/30)

[【JAVA】/▶ 09.Oracle](http://devhome.tistory.com/category/%E3%80%90JAVA%E3%80%91/%E2%96%B6%2009.Oracle) 2009.07.10 10:59

오라클에 BLOB 데이터를 입력하는 방법을 소개하고자 한다.   
  
예를 들어 TEST\_TBL 이라는 테이블이 존재한다는 가정하에 설명하겠다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **PK/Nullable** | **Type** | **Default** | **Comment** |
| REC\_KEY | PK | Number(8) |  | 키 |
| USER\_ID | Not Null | Varchar(50) |  | 사용자 아이디 |
| USER\_PW | Not Null | Varchar(50) |  | 사용자 비밀번호 |
| PHOTO |  | BLOB |  | 사용자 사진 |

[?](http://devhome.tistory.com/30?category=108315)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29 | // 오라클 드라이버 로딩  Class.forName("oracle.jdbc.driver.OracleDriver");    // 오라클 연결  Connection conn = DriverManager.getConnection("jdbc:oracle:thin:@127.0.0.1:1521:Service", "User", "PassWord");    // 파일 정보  File file = new File("파일경로");  InputStream is = new FileInputStream(file);  int fileSize = (int)file.length();    // 입력쿼리 구성  String insertSql = "INSERT INTO TEST\_TBL VALUES( 1, 'User', 'Password', ? )";  PreparedStatement psmt = conn.prepareStatement(insertSql);    // [스트림을 이용하는 방법]  psmt.setsetBinaryStream(1, is, fileSize);    // [byte배열을 이용하는 방법]  byte[] buffer = new byte[fileSize];  is.read(buffer);  psmt.setBytes(1, buffer);    // 입력  int insertCount = psmt.executeUpdate();  psmt.close();    // 연결해제  conn.close(); |

---------------------------------------------------------------------------------------------

[**오라클(Oracle) CLOB 데이터 입력 - JDBC**](http://devhome.tistory.com/17)

[【JAVA】/▶ 09.Oracle](http://devhome.tistory.com/category/%E3%80%90JAVA%E3%80%91/%E2%96%B6%2009.Oracle) 2009.04.07 14:23

JDBC로 CLOB 필드를 처리하는데 있어서 많이 쉬워졌다.  
   
간단하게 CLOB 필드에 데이터를 넣는 방법은 아래 코드를 참조하기 바란다.

[?](http://devhome.tistory.com/17?category=108315)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | // Make Insert Query  String strQuery = "INSERT INTO TEST\_TBL VALUES( ?, ?, ?, ?)";  PreparedStatement pstmt = con.prepareStatement(strQuery);  pstmt.setInt(1, 1);  pstmt.setString(2, "TITLE INFO");  pstmt.setString(3, "AUTHOR INFO");  pstmt.setString(4, "CLOB DATA");    // Insert Row  int nRowCnt = pstmt.executeUpdate();  pstmt.close(); |

**=======================================================================================================**  
  
오라클(Oracle)의 CLOB 데이터를 입력하기 위해서는 일반 Field에 데이터를 넣는 방식과는 다른 방법이 필요하다.  
  
**CLOB 데이터는 다른 Field의 데이터와 함께 넣을 수 없으면 오직 혼자서만 입력이 가능하다.**  
  
오라클에 CLOB 데이터를 입력하기 위해서는 다음과 같은 순서로 작업을 해야한다.  
  
예를 들어 TEST\_TBL 이라는 테이블이 존재한다는 가정하에 설명하겠다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **PK/Nullable** | **Type** | **Default** | **Comment** |
| REC\_KEY | PK | Number(8) |  | 키 |
| TITLE | Not Null | Varchar(200) |  | 제목 |
| AUTHOR | Not Null | Varchar(100) |  | 저자 |
| CONTENTS |  | Clob |  | 내용 |

**1. 사전에 CLOB을 제외한 Field 데이터를 가지는 Row를 입력한다.**  
  
 - Insert Into TEST\_TBL Valuse( 1, 'XXXXX', 'XXXXX', Empty\_Clob() )  
  
  
**2. 해당 Row에 Row Lock을 걸어준다.**  
- Select CONTENTS From TEST\_TBL Where REC\_KEY = 1 For Update  
  
  
**3. 해당 Field에 CLOB 데이터를 입력한다.**  
  
다음은 위의 CLOB 데이터 입력 순서를 적용한 예제 소스이다.

[?](http://devhome.tistory.com/17?category=108315)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63 | import java.io.CharArrayReader;  import java.io.IOException;  import java.io.Reader;  import java.io.Writer;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.sql.SQLException;  import oracle.jdbc.driver.OracleResultSet;  import oracle.sql.CLOB;    public class OracleClob {      public static void main(String[] args) throws IOException, SQLException, ClassNotFoundException {          // Connect Oracle          Class.forName("oracle.jdbc.driver.OracleDriver");          Connection con = DriverManager.getConnection("jdbc:oracle:thin:@127.0.0.1:1521:Service", "User", "PassWord");            // Auto Commit false          con.setAutoCommit(false);            // Make Insert Query          String strQuery = "INSERT INTO TEST\_TBL VALUES( ?, ?, ?, EMPTY\_CLOB() )";          PreparedStatement pstmt = con.prepareStatement(strQuery);          pstmt.setInt(1, 1);          pstmt.setString(2, "XXXXX");          pstmt.setString(3, "XXXXX");            // Insert Row          int nRowCnt = pstmt.executeUpdate();          pstmt.close();            if( nRowCnt == 1 ) {              // Make Select Query & Row Lock              strQuery = "SELECT CONTENTS FROM TEST\_TBL WHERE REC\_KEY = ? FOR UPDATE";              pstmt = con.prepareStatement(strQuery);              pstmt.setInt(1, 1);              ResultSet rs = pstmt.executeQuery();                // Write CLOB Data              String strCLOB = "XXXXX";              if( rs.next() ) {                  CLOB clob = ((OracleResultSet)rs).getCLOB("CONTENTS");                  Writer writer = clob.getCharacterOutputStream();                  Reader reader = new CharArrayReader(strCLOB.toCharArray());                  char[] buffer = new char[1024];                  int read = 0;                    while ((read = reader.read(buffer, 0, 1024)) != -1) {                      writer.write(buffer, 0, read);                  }                  reader.close();                  writer.close();              }                // Commit              con.commit();              con.setAutoCommit(true);              rs.close();          }          con.close();      }  } |

출처: <http://devhome.tistory.com/17?category=108315> [미주엘의 개발이야기]