

Álvaro García Fuentes
Horas Libre Configuración
Desarrollo de Aplicaciones Web

UD3 A4 Actualización mediante INSERT, UPDATE o DELETE

1. Indica verdadero/falso

1. **Un objeto ResultSet predeterminado no es actualizable y tiene un cursor que sólo se mueve hacia adelante.**

Verdadero

2. **Todos los objetos ResultSet pueden actualizarse y/o moverse (moverse del final al principio con el cursor).**

Falso, los objetos ResultSet no se pueden actualizar y su cursor sólo se mueve desde el principio hacia el final.

3. **En un objeto ResultSet scrollable, el cursor puede moverse delante y atrás.**

Verdadero.

4. **Los métodos update de ResultSet sólo permiten identificar la columna por su nombre y por su número (0..totalColumnas-1).**

Verdadero.

5. **Los métodos update devuelven un booleano para indicar o no que la actualización se ha realizado.**

Falso, los métodos update son todos void.

6. **Los métodos de actualización "update*" se utilizan para actualizar los valores de columna en la fila actual o la fila de inserción.**

Verdadero.

7. **Los métodos de actualización "update*" no actualizan la base de datos subyacente; en su lugar, se llama a los métodos updateRow o insertRow para actualizar la base de datos.**

Verdadero.

8. **El método updateRow actualiza la tabla de origen de datos de la que se derivó el ResultSet.**

Verdadero.

2. Indica los campos del ResultSet y su significado.

CLOSE_CURSORS_AT_COMMIT: al abrir un ResultSet con este campo se cerrará nada más realizar la primera transacción.

CONCUR_READ_ONLY: indica que un ResultSet no será actualizado.

CONCUR_UPDATABLE: indica que un ResultSet se podrá actualizar.

FEETCH_FORWARD: indica que las filas del ResultSet serán procesadas del principio al fin.

FECTH_REVERSE: indica que las filas del ResultSet serán procesadas de foma inversa, es decir, desde el final al principio.

FECTH_UNKNOWN: indica que el orden a procesar las filas es desconocido.

HOLD_CURSORS_OVER_COMMIT: indica que el ResultSet permanecerá abierto después de que se realice la transacción actual.

TYPE_FORWARD_ONLY: indica el tipo de un ResultSet cuyo cursor se mueve sólo de principio a fin.

TYPE_SCROLL_INSENSITIVE: indica el tipo de un ResultSet que es scrollable pero generalmente no sensible a cambios en los datos que subyacen al ResultSet.

TYPE_SCROLL_SENSITIVE: indica el tipo de un ResultSet que es scrollable y además sensible a cambios en los datos que subyacen al ResultSet.

3. Indica el tipo de los objetos ResultSet obtenidos a raíz de createStatement() sin parámetros.

Por defecto con createStatement se producen objetos ResultSet que no son scrollables ni actualizables.

4. Además del método next, existen otros métodos para mover el cursor. Indícalos.

Los métodos para mover el cursor además de next() son: absolute(), afterLast(), beforeFrist(), first(), last(), moveToCurrentRow(), moveToInsertRow(), previous() y relative().

5. Indica si al igual que los métodos getxxx, los métodos updatexxx están sobrecargados.

Sí, ResultSet tiene varios métodos update con el mismo nombre pero distintos parámetros de entrada.

6. Implementa la clase Reverso. Muestra las filas de una tabla de abajo a arriba.

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.SQLTimeoutException;
import java.sql.Statement;

public class Reverso {

    public static void main(String[] args){

        String db = "HLC";
        String login = "alvaro";
        String password = "macarrones";
        String url = "jdbc:mysql://127.0.0.1:3306/" + db;

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection conexion = DriverManager.getConnection(url, login, password);

            if( conexion != null ) {
                Statement st = conexion.createStatement( ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.FETCH_REVERSE );
                System.out.println( "Conexion a base de datos " + db + " correcta." );
                ResultSet rs = st.executeQuery("SELECT * FROM ALUMNOS;");
                rs.afterLast();
                System.out.println("tabla ALUMNOS");
                while( rs.previous() ) {
                    System.out.println( "Nombre: " + rs.getString("nombre")
                                         + " Curso: " + rs.getInt("curso")
                                         + " Nota media: " + rs.getInt("notaMedia") );
                }
            } else {
                System.out.println("Conexion fallida.");
            }

        } catch ( SQLTimeoutException e ) { System.out.print("Error de tiempo de conexion."); }
        catch ( SQLException e ) { e.printStackTrace(); }
        catch ( ClassNotFoundException e ) { e.printStackTrace(); }
        catch ( Exception e ) { e.printStackTrace(); }

    }
}
```

```
Conexion a base de datos HLC correcta.
tabla ALUMNOS
Nombre: pepe Curso: 1 Nota media: 8
Nombre: maria Curso: 2 Nota media: 7
Nombre: fernando Curso: 1 Nota media: 9
Nombre: antonio Curso: 3 Nota media: 5
Nombre: ana Curso: 2 Nota media: 6
```

7. Implementa la clase NotaMedia. Muestra el nombre y nota media de cada uno de los alumnos. A posteriori muestra a los alumnos con la mejor y con la peor nota media. Para ello crea una consulta que devuelva los alumnos ordenados por su nota media. Accede entonces al primer y último elemento.

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.SQLTimeoutException;
import java.sql.Statement;

public class NotaMedia {

    public static void main(String[] args){

        String db = "HLC";
        String login = "alvaro";
        String password = "macarrones";
        String url = "jdbc:mysql://127.0.0.1:3306/" + db;
        String query = "SELECT * FROM ALUMNOS ORDER BY notaMedia;";

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection conexion = DriverManager.getConnection(url, login, password);

            if( conexion != null ) {
                Statement st = conexion.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.FETCH_UNKNOWN);
                System.out.println( "Conexion a base de datos " + db + " correcta." );
                ResultSet rs = st.executeQuery(query);

                System.out.println("tabla ALUMNOS");
                while( rs.next() ) {
                    String nombre = rs.getString("nombre");
                    int notaMedia = rs.getInt("notaMedia");
                    int curso = rs.getInt("curso");
                    System.out.println("Nombre: " + nombre + ", curso: " + curso + ", nota media: " + notaMedia);
                }

                rs = st.executeQuery(query);
                rs.absolute(1);
                System.out.println( "Alumno con peor nota: " + rs.getString("nombre") + ", nota media "
                    + rs.getFloat("notaMedia") + " en curso " + rs.getInt("curso") );
                rs.last();
                System.out.println( "Alumno mejor nota: " + rs.getString("nombre") + ", nota media "
                    + rs.getFloat("notaMedia") + " en curso " + rs.getInt("curso") );

            } else
                System.out.println("Conexion fallida.");

        } catch ( SQLTimeoutException e ) { System.out.print("Error de tiempo de conexion."); }
        catch ( SQLException e ) { e.printStackTrace(); }
        catch ( ClassNotFoundException e ) { e.printStackTrace(); }
        catch ( Exception e ) { e.printStackTrace(); }
    }
}
```

```
Conexion a base de datos HLC correcta.
tabla ALUMNOS
Nombre: antonio, curso: 3, nota media: 5
Nombre: ana, curso: 2, nota media: 6
Nombre: maria, curso: 2, nota media: 7
Nombre: pepe, curso: 1, nota media: 8
Nombre: fernando, curso: 1, nota media: 9
Alumno con peor nota: antonio, nota media 5.0 en curso 3
Alumno mejor nota: fernando, nota media 9.0 en curso 1
```

8. Implementa la clase Delete que elimine los alumnos de un curso determinado.

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.SQLTimeoutException;
import java.sql.Statement;

public class Delete {

    public static void main(String[] args){

        String db = "HLC";
        String login = "alvaro";
        String password_ = "macarrones";
        String url = "jdbc:mysql://127.0.0.1:3306/" + db;
        String query = "SELECT * FROM ALUMNOS ORDER BY notaMedia;";

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection conexion = DriverManager.getConnection(url, login, password_);

            if( conexion != null ) {
                Statement st = conexion.createStatement();
                System.out.println( "Conexion a base de datos " + db + " correcta." );
                ResultSet rs = st.executeQuery(query);

                System.out.println("tabla ALUMNOS");
                while( rs.next() ) {
                    System.out.println( "Nombre: " + rs.getString("nombre")
                                         + ", curso: " + rs.getInt("curso")
                                         + ", nota media: " + rs.getInt("notaMedia") );
                }

                System.out.println( "Borramos los alumnos de 3º curso." );

                st = conexion.createStatement();

                int nEliminados = st.executeUpdate("DELETE FROM ALUMNOS WHERE curso=3;");
                System.out.println( "Alumnos eliminados: " + nEliminados );
                st = conexion.createStatement();

                System.out.println("tabla ALUMNOS");
                rs = st.executeQuery(query);
                while( rs.next() ) {
                    System.out.println( "Nombre: " + rs.getString("nombre")
                                         + ", curso: " + rs.getInt("curso")
                                         + ", nota media: " + rs.getInt("notaMedia") );
                }
            } else {
                System.out.println("Conexion fallida.");
            }
        } catch ( SQLTimeoutException e ) { System.out.print("Error de tiempo de conexion."); }
        catch ( SQLException e ) { e.printStackTrace(); }
        catch ( ClassNotFoundException e ) { e.printStackTrace(); }
        catch ( Exception e ) { e.printStackTrace(); }
    }
}
```

Álvaro García Fuentes
Horas Libre Configuración
Desarrollo de Aplicaciones Web

```
Conexion a base de datos HLC correcta.
tabla ALUMNOS
Nombre: antonio, curso: 3, nota media: 5
Nombre: ana, curso: 2, nota media: 6
Nombre: maria, curso: 2, nota media: 7
Nombre: pepe, curso: 1, nota media: 8
Nombre: fernando, curso: 1, nota media: 9
Borramos los alumnos de 3º curso.
Alumnos eliminados: 1
tabla ALUMNOS
Nombre: ana, curso: 2, nota media: 6
Nombre: maria, curso: 2, nota media: 7
Nombre: pepe, curso: 1, nota media: 8
Nombre: fernando, curso: 1, nota media: 9
```