

Álvaro García Fuentes
Horas Libre Condiguración
Desarrollo de Aplicaciones Web

UD2 A6 Transacciones

1. Transacción. Indica qué es.

Es un conjunto de operaciones sobre una base de datos que se deben ejecutar como una unidad.

2. Cuando una conexión se crea en JDBC trabaja en modo auto-commit (auto entrega). Indica qué significa.

Un objeto Connection por defecto realiza automáticamente cada operación sobre la base de datos. Esto significa que cada vez que se ejecuta una instrucción, se refleja en la base de datos y no puede ser deshecha. Por defecto está habilitado el modo auto-commit en la conexión.

Para iniciar una transacción deshabilitamos el modo auto-commit mediante el método `setAutoCommit(false)`.

3. Indica los dos métodos necesarios para agrupar las sentencias SQL en una transacción.

Son `connection.setAutoCommit(false)` y `connection.commit()`.

4. Entrega un ejemplo de transacción original donde mantengas la consistencia de los datos tras varias actualizaciones.

```
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 Transacciones {
    public static void main(String[] args) {
        String url = "jdbc:mysql://127.0.0.1:3306/HLC";
        String login = "alvaro";
        String password = "macarrones";
        try {
            Connection conexion = DriverManager.getConnection(url, login, password);
            conexion.setAutoCommit(false);

            // primero consultamos la tabla alumnos para ver su estado inicial
            Statement st = conexion.createStatement();
            ResultSet rs = st.executeQuery("SELECT nombre, notaMedia, curso FROM ALUMNOS");
            System.out.println("tabla ALUMNOS");
            while(rs.next()) {
                System.out.println( "Nombre: " + rs.getString("nombre")
                                    + " Nota media: " + rs.getInt("notaMedia")
                                    + " Curso: " + rs.getInt("curso"));
            }

            // Hacemos las actualizaciones con una transacción
            st = conexion.createStatement();
            st.executeUpdate("INSERT INTO ALUMNOS(nombre,notaMedia,curso) VALUES('maria',8,1)");
            st.executeUpdate("INSERT INTO ALUMNOS(nombre,notaMedia,curso) VALUES('pepe',7,2)");
            st.executeUpdate("INSERT INTO ALUMNOS(nombre,notaMedia,curso) VALUES('ana',9,3)");
            st.executeUpdate("INSERT INTO ALUMNOS(nombre,notaMedia,curso) VALUES('fernando',6,2)");
            conexion.commit();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Álvaro García Fuentes
Horas Libre Configuración
Desarrollo de Aplicaciones Web

```
// Volvemos a consultar la tabla alumnos para ver su estado final
st = conexion.createStatement();
rs = st.executeQuery("SELECT nombre, notaMedia, curso FROM ALUMNOS");
System.out.println("tabla ALUMNOS");
while(rs.next()) {
    System.out.println( "Nombre: " + rs.getString("nombre")
                        + " Nota media: " + rs.getInt("notaMedia")
                        + " Curso: " + rs.getInt("curso"));
}

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

Y el resultado es el siguiente:

```
tabla ALUMNOS
tabla ALUMNOS
Nombre: ana Nota media: 9 Curso: 3
Nombre: fernando Nota media: 6 Curso: 2
Nombre: maria Nota media: 8 Curso: 1
Nombre: pepe Nota media: 7 Curso: 2
```

Al principio la tabla ALUMNOS está vacía, y en la segunda consulta después de usar la transacción ya no lo está.