Transacciones

Antes de empezar con las transacciones, vamos a ver una consulta sencilla de manipulación de los datos: insertar un registro

Insertar un registro en una tabla

Por ejemplo, vamos a ver cómo podemos insertar un paciente nuevo en la base de datos Hospital:

```
Conexion con = new Conexion();
Connection conexion = con.conexionMySQL();
PreparedStatement sentencia;
try {
   String sql = "INSERT into paciente (NumHistorial, Nombre, Apellidos) "
         + "VALUES (?,?,?)";
   sentencia = conexion.prepareStatement(sql);
   //Sustitumos por datos de ejemplo: Historia 100, nobmre Paciente100, apellids 100
   sentencia.setInt(1, 100);
   sentencia.setString(2, "Paciente100");
   sentencia.setString(3, "Apellidos100");
   //ejecutamos la inserción o actualización de los registros de la tabla
   sentencia.executeUpdate();
   System.out.println("Insertado paciente.");
   sentencia.close();
} catch (SQLException ex) {
   System.err.println("Error SQL. " + ex.getMessage());
}
if (conexion != null) {
   try {
      conexion.close();
   } catch (SQLException ex) {
      System.err.println("Error al cerrar la conexion. " + ex.getMessage());
   }
}
```

Transacciones. Repasando sus características:

Una transacción es un conjunto de operaciones que deben ejecutarse como si fuese una unidad, como un todo, de manera aislada al resto de operaciones que se puedan estar realizando en paralelo con otro proceso sobre los mismos datos.

Las operaciones que forman parte de la transacción se ejecutan todas completamente, como un bloque, confirmando todos los cambios en la base de datos. De manera que, si hay un error, las operaciones puedan no ejecutarse o deshacerse para dejar la base de datos en el estado inicial, el que tenía antes de comenzar la transacción.

Las características de una transacción eran ACID:

- Atomic
- Consistent
- Isolated
- Durable.

Sin lugar a duda, las operaciones estrellas de los gestores son las transacciones. Por eso, la mayoría las incorpora en el propio gestor, por ejemplo:

- ORACLE: Las tiene habilitadas por defecto, no se confirma ningún cambio en la base de datos hasta que se ejecuta COMMIT.
- MySQL: Las tiene deshabilitadas por defecto. De manera que los cambios realizados por cualquier sentencia se confirman automáticamente, sin necesidad de ejecutar COMMIT.

JDBC proporciona una interfaz común para todas las bases de datos porque, aunque podemos controlar con SQL las transacciones directamente, la aplicación solo funciona para una bd particular.

Código de una transacción:

```
START TRANSACTION
    Operación 1 sobre la bd
    …
    Operación n sobre la bd.
COMMIT
```

La forma de abortar o cancelar una transacción, volviendo al estado inicial la bd, el que tenía antes de efectuar cualquier operación de la transacción, tenemos que usar la sentencia ROLLBACK.

La interfaz Connection tiene los métodos para hacer las operaciones relacionadas con las transacciones.

void setAutoCommit (boolean autoCommit)	autoCommit=false Inicia transacción (equivale a START TRANSACTION

void commit()	Se puede hacer con autoCommit =true en el método setAutCommit(true)
void rollback()	Descarta cambios Se hace tras cualquier excepción SQLException

Ejemplo de transacción: Insertar un médico nuevo en la base de datos hospital, cuyo CodIdentificacion será 10 veces más alto que el máximo que haya almacenado.

```
Conexion con = new Conexion();
Connection conexion = con.conexionMySQL();
PreparedStatement sentencia;
try {
   //voy a controlar commit, para ello setAtocommit(false)
   conexion.setAutoCommit(false);
   //recupero el identificador más alto
   Statement st = conexion.createStatement();
   ResultSet rs = st.executeQuery("select max(codidentificacion) id from medico");
   rs.next();
   int maxid = rs.getInt(1);
   //Insertamos el registro del médico añadiendo 10 al maxid
   String sql = "INSERT into medico (CodIdentificacion, Nombre, Apellidos, Especialidad) "
         + "VALUES (?,?,?,?)";
   sentencia = conexion.prepareStatement(sql);
   sentencia.setInt(1, maxid+10);
   sentencia.setString(2, "Medico"+(maxid+10));
   sentencia.setString(3, "Apellidos"+(maxid+10));
   sentencia.setString(4, "Medicina Familiar");
   //ejecutamos la inserción o actualización de los registros de la tabla
   sentencia.executeUpdate();
   System.out.println("Insertado médico.");
   //ahora hacemos el commit
   conexion.commit();
   sentencia.close();
} catch (SQLException ex) {
   System.err.println("Error SQL. " + ex.getMessage());
   try{
      System.out.println("ROLLBACK. Hubo un error.");
      conexion.rollback();
   }catch (SQLException e){
      System.err.println("Error en la transacción. " + e.getMessage());
   }
}
if (conexion != null) {
   try {
      conexion.setAutoCommit(true); //cambio el autocommit. Puede omitirse pq cierro
```

```
            conexion.close();
        } catch (SQLException ex) {
            System.err.println("Error al cerrar la conexion. " + ex.getMessage());
        }
    }
```