UF6: POO. Introducción a la persistencia en BD

DESARROLLO DE APLICACIONES

ABRIL DE 2023

Innovación en Formación Profesional

De:

Planeta Formación y Universidades

Contenido

1. BBDD Relacionales. Introducción.
2. BBDD Relacionales: Establecimiento de conexiones.
3. BBDD Relacionales: Recuperación de información.
4. BBDD Relacionales: Manipulación de la información.
5. BBDD Objeto-Relacionales. Introducción.
6. BBDD Objeto-Relacionales: Establecimiento de conexiones.
7. BBDD Objeto-Relacionales: Recuperación de información.
8. BBDD Objeto-Relacionales: Manipulación de la información.
9. BBDD Objeto. Introducción a las bases de datos orientadas a objetos.
10. BBDD Objeto: Características
11. BBDD Objeto: Modelo de datos orientado a objetos: relaciones, integridad, UML.
12. BBDD Objeto: El modelo estándar ODMG
13. Prototipos y productos comerciales de SGBDOO.

BBDD Relacionales: Introducción

Una BD relacional almacena los datos en tablas, que a su vez se componen de una serie de filas las cuales tienen los mismos elementos (columnas).

Se denomina relacional porque proporciona acceso a datos que están relacionados entre sí.

Un SGBD es el proceso responsable de manejar, almacenar y recuperar los datos de una BBDD.

Java tiene un API JDBC que ofrece acceso a BD relacionales, pudiendo realizar consultas SQL, recuperar datos, realizar cambios, etc.

BBDD Relacionales: Establecer conexiones

Para trabajar con JDBC necesitamos:

- Versión de Java (las clases para trabajar con BBDD están en el paquete java.sql)
- Una Base de datos, por supuesto. En este caso utilizaremos MySQL.
- Los drivers necesarios para conectarse con la BBDD utilizada.

Para conectar a MySQL tenemos tres opciones:
- Mediante consola (comando psql)
- Mediante herramienta gráfica (PostgreSQL incluye pgAdmin).
- Mediante las clases del API JDBC.

Antes de trabajar con una BBDD debemos establecer una conexión con la misma. JDBC se conecta utilizando una de estas dos clases:
- DriverManager. Es la forma más sencilla. Se realiza a través de una dirección URL.
- DataSource. Hace que los detalles de la BBDD sean transparentes a la aplicación.

```
try {
  Connection connexion = DriverManager.getConnection("jdbc:mysql://localhost:3306/test", "username", "passwd");
  System.out.println("Conexión establecida");
}
catch (Exception e){
  System.out.println("Error al conectarse a la BBDD");
}
```

BBDD Relacionales: Establecer conexiones

La URL: “jdbc:mysql://localhost:3306/test", ¿qué significa?

jdbc:mysql: protocolo jdbc y SGBD MySQL al que se conecta.

localhost: Dirección de la máquina donde reside la BBDD

3306: Puerto donde escucha por defecto la BBDD

test: Nombre de la BBDD

El método *getConnection* además de la URL necesita el usuario con el que se conecta a la BBDD y la password.

El método retorna un objeto de tipo *Connection*, que ya tiene una conexión con la BBDD y podemos empezar a hacer operaciones contra la BBDD.

Cuando JDBC encuentra un error al trabajar con una BBDD lanza una *SQLException*. Este objeto contiene mucha información para determinar el origen del error.

BBDD Relacionales: Recuperación de información

Para poder ejecutar una consulta SQL en una BBDD tenemos que trabajar con un objeto de tipo *Statement*.

Existen tres tipos de objetos *Statement*:

- *Statement:* Sirve para enviar órdenes SQL a la BBDD sin parámetros.
- *PreparedStatement:* hereda de *Statement*, ejecuta comandos SQL con o sin parámetros ya precompilados. Es más eficiente que *Statement*.
- *CallableStatement:* Hereda de *PreparedStatement*, llama a procedimientos almacenados en BBDD.

Una vez realizada la conexión y creado el objeto Statement/PreparedStatement/CallableStatement se pueden llamar a tres métodos diferentes para ejecutar sentencias SQL:

executeQuery: Se utiliza para ejecutar sentencias SELECT. Retorna un *ResultSet*.

executeUpdate: Se puede utilizar en sentencias INSERT, UPDATE o DELETE.

execute: Utilizado en sentencias que devuelven más de un *ResultSet*.

Se recomienda cerrar los objetos *statement* mediante el método *close()*.

Para obtener los datos de una tabla contenidos en un ResultSet debemos recorrerlo y obtener los valores a partir de los nombres de los campos (columnas) o bien a través de métodos que llaman a los campos por el orden que ocupan en la sentencia SQL.

BBDD Relacionales: Manipulación de la información.

La modificación de una tabla en una BBDD es similar a la ejecución de otras sentencias como las inserciones y borrados en tablas. Únicamente cambia la sintaxis SQL.

```
Statement stmt = null;
Stmt = con.createStatement();
Stmt.executeUpdate("update departamento set nombre='Informática' where idDepto=1  ");
…
```

En Java se pueden crear *ResultSet* bidireccionales y actualizables. Para ello hay que definir el *ResultSet* con la propiedad TYPE_SCROLL_SENSITIVE., que hace que el objeto *ResultSet* pueda moverse bidireccionalmente de forma relativa a su posición actual y, por otro lado, la propiedad CONCUR_UPDATABLE hace que se puedan modificar los datos del cursos y estos se repliquen en la BBDD. Hasta que no se invoca el método *ResultSet.updateRow* o se actualizará la BBDD.

BBDD Objetos-Relacionales: Establecer conexiones

Es un modelo que intenta fusionar la orientación a objetos con el modelo de BBDD relacional.

Una estrecha integración entre los dos modelo facilitaría el desarrollo evitando realizar una traslación de los datos de las tablas a objetos.

Utilizaremos para nuestros ejemplos: PostgreSQL.

Para conectar a PostgreSQL (al igual que con MySQL) tenemos tres opciones:

- Mediante consola (comando psql)
- Mediante herramienta gráfica (PostgreSQL incluye pgAdmin).
- Mediante las clases del API JDBC

Si accedemos mediante el API JDBC:

```
Connection connection = null;
connection = DriverManager.getConnection("jdbc:postgresql://192.168.0.26:5432/ejemploDB", "userName",
"passwd");
```

BBDD Objetos-Relacionales: Recuperación de información

```
try {

  Empleado objeEmpleado = new Empleado();
  objeEmpleado.nombre = resultset.getString("nombre");
  objeEmpleado.apellido = resultset.getString("apellido");
  objeEmpleado.telefono = resultset.getString("telefono");
  objeEmpleado.num_empleado = resultset.getInt("num_empleado");
}
catch (SQLException ex) {
  ex.printStackTrace();
}
```

apellido	nombre	telefono	num_empleado
Elorduy	Gorka	888-888-888	1
Gómez	Pepe	777-777-777	2
Pérez	Luis	555-555-555	3

BBDD Objetos-Relacionales: Manipulación de información

La modificación de una tabla en una BBDD es similar a la ejecución de otras sentencias como las inserciones y borrados en tablas. Únicamente cambia la sintaxis SQL.

```
Statement stmt = null;
Stmt = con.createStatement();
Stmt.executeUpdate("update departamento set nombre='Informática' where idDepto=1  ");
…
```

BBDD Objetos: Introducción

Las BD relacionales presentan algunas deficiencias cuando se trata de aplicaciones más complejas o sofisticadas. La estructura de objetos suele ser más compleja y se necesitan tipos nuevos para almacenar información.

almacena los datos en tablas, que a su vez se componen de una serie de filas las cuales tienen los mismos elementos (columnas).

Se denomina relacional porque proporciona acceso a datos que están relacionados entre sí.

Un SGBD es el proceso responsable de manejar, almacenar y recuperar los datos de una BBDD.

Java tiene un API JDBC que ofrece acceso a BD relacionales, pudiendo realizar consultas SQL, recuperar datos, realizar cambios, etc.

BBDD Objetos: características

- Ofrecen flexibilidad y no están limitadas por los tipos de datos y los lenguajes de consulta.
- Proporcionan potencia al diseñador al permitirle especificar tanto la estructura de objetos complejos, como las operaciones que se pueden aplicar sobre dichos objetos.
- En línea con los lenguajes de POO.
- Debe incorporar las características de la POO: Encapsulación, identidad de objetos, extensibilidad, etc.
- Opcionalidad para añadir herencia múltiple, transacciones y versiones, chequeo de tipos e inferencia de la distribución.
- Concurrencia
- Recuperación
- Facilidad de consultas
- Herencia múltiple
- Diseño de transacciones y versiones.

Algunos de estas características son obligatorias, en tanto que otras son opcionales.

BBDD Objetos: El modelo de datos orientado a objetos

Relaciones.

Las BBDD relacionales representan las relaciones mediante claves ajenas. Por el contrario las BDOO implementan sus relaciones incluyendo en cada objeto los identificadores de los objetos con los que se relaciona. Un identificador de objeto es un atributo interno que posee cada objeto.

Integridad.

Para que las relaciones funciones en una BBDDOO pura, los identificadores de los objetos deben corresponderse en ambos extremos de la relación.

UML (Unified Modeling Language).

Es una de las notaciones existentes para diseñar esquemas conceptuales de BBDD orientadas a objetos.

BBDD Objetos: El modelo estándar ODMG

Un grupo de representantes de la industria de las bases de datos formaron el ODMG (Object Database Management Group) con el propósito de definir estándares para los SGBD orientados a objetos.

Este grupo propuso un modelo estándar para la semántica de los objetos de una base de datos.

Los principales componentes de la arquitectura ODMG para un SGBD orientado a objetos son los siguientes:

- Modelo de objetos: permite la portabilidad entre los sistemas que lo soportan.

- Lenguaje de definición de objetos (ODL). Es un lenguaje de especificación para definir tipos de objetos para sistemas compatibles con ODMG. Es el equivalente al DDL de los SGBD tradicionales.

- Lenguaje de consulta de objetos (OQL). Tipo SQL permite realizar consultas sobre BBDD orientadas a objetos. Basado en SQL-92.

- Conexión con los lenguajes C++, Smalltalk y Java.

BBDD Objetos: Prototipos y productos comerciales de SGBDOO

- ObjectDB
- Zope Object Database
- Object Database++
- ObjectStore
- Wakanda
- GemStone/S
- db40
- EyeDB
- Intersystems Cache
- Objectivity/DB