CREACIÓN DE NUEVAS CONSULTAS

Ya que las consultas son parte fundamental de este proyecto, es necesario dedicar un apartado para describir la estructura que tienen en general todas las consultas, los métodos que se pueden utilizar por herencia de la clase Infxclass.java y los pasos a seguir para generar consultas en un futuro. Las consultas que están predefinidas en nuestra aplicación son sólo una muestra de lo que se puede lograr realizando las diferentes relaciones producto de modelo entidad-relación que se explicó en el capítulo 3. Dichas consultas han sido definidas en el capítulo 4.

Herencia de Infxclass

La clase Queries.java es una extensión de la clase Infxclass. Esto significa que todos los atributos así como los métodos descritos en Infxclass, pueden ser accesados y utilizados respectivamente por un objeto tipo Queries.

La clase Infxclass es un paquete que prevé de una conexión a una base de datos, programada por Nidia Posada. En esta clase existen métodos como open que se encarga de abrir la conexión con la base de datos, executeQuery que recibe una cadena representando la consulta en SQL o executeUpdate, executeInsert, etc.

La clase Queries extiende Infxclass, porque todo método de esta clase tiene una consulta a la base de datos relacionada, y por lo mismo era más sencillo y funcional

extenderla que referenciarla. El acceso a los métodos de la Super clase se genera de manera directa, a diferencia de cuando esta referenciada.

Cabe señalar que todas las consultas prediseñadas, siempre reciben algún objeto que hace referencia a lo que se quiere consultar, y también siempre regresan un objeto que por lo regular son vectores, cadenas de caracteres u objetos de los que han sido descritos al principio del capítulo 4.

Pasos para generar una consulta

- Se recomienda al desarrollador, inicialmente analizar el esquema entidad relación, y generar la relación entre tablas de manera rudimentaria en papel.
- Después se recomienda probar la consulta en el ambiente del DBMS que en este caso se ha usado "dbaccess" de Informix. Con esto se podrá observar si el resultado concuerda con el deseado.
- Ya que se está seguro que se obtiene el resultado deseado de la consulta se analizan las clases necesarias que deben ser involucradas con la consulta, para de tal manera, almacenar los resultados de la consulta en un objeto el cual puede ser más fácil de manipular.
- Ya que se identificó el tipo de objeto que la consulta generará se decide si es lo único necesario para poder desplegar el resultado. Por ejemplo: Quiero obtener todos los nombres de las escuelas que se encuentran en las localidades a menos de "x" kilómetros a la redonda del cráter del volcán. Si lo único que necesito son los nombres, con un vector de cadenas de

caracteres es suficiente, pero si quiero toda la información de todas las escuelas, es pertinente generar objetos de tipo Escuela temporales y almacenarlos en un vector. Como resultado de esta consulta obtendremos un vector de objetos tipo Escuela, el cual es más sencillo manipular, que si se obtuvieran todos los datos por separado.

- Ya se tiene definido el objeto que se va a utilizar como resultado. Ahora viene la parte de la conexión a la base de datos. Como ya se menciono la clase Infxclass ya tiene sus métodos bien estructurados, y por lo tanto sólo se tiene que mandar a llamar el método que genera el Query incluyendo la cadena de texto que sea la representación misma de la consulta. Por ejemplo: tenemos el nombre de una localidad y queremos obtener todos los datos de ella, para generar un objeto de tipo localidad, el método seria:

```
Public Localidad nuevaLocalidad(String nombre_localidad)
{
this.queryResult = this.executeQuery(“ select * from localidad where nombre_localidad = ´”+nombre_localidad+”´,”);
try
{
while(queryResult.next())
{
localidadNueva.setNombre(“nombre_localidad”)
localidadNueva.setClaveLocalidad(queryResult.getInt(“clave_localidad”));
.....
return localidadNueva;

}
```

- Como se observa en el segmento de código anterior, existe un atributo de tipo ResultSet el cual se llama queryResults . Este atributo es donde se guarda la información sin tratamiento obtenida de alguna consulta a la base de datos. En la clase ResultSet hay métodos que sirven para obtener en representación de objeto todos los tipos de datos que se pueden declarar en una base de datos. Están el getString, getFloat, getInt, etc. Con la siguiente instrucción se puede obtener un vector de Strings de la columna de la tabla llamada nombre:

```
while(queryResults.next())
{

    vectorStrings.addElement(queryResults.getString("nombre"));

}
```

- Ya que se tiene la consulta diseñada, lo único que hay que hacer es insertar la consulta en el Servlet. El procedimiento para agregar la consulta es muy sencillo. Tomando el código que se enlisto arriba generaremos las líneas de código necesarias para introducir una consulta en nuestro Servlet.

```
<%
Localidad  nueva_localidad = new Localidad();
Queries temporal = new Queries();
nueva_localidad = temporal.nuevaLocalidad(name);
//          Para imprimir los datos de la localidad
out.println(“ El nombre es: ”+nueva_localidad.getNombre());
out.println(“ Su clave es: “+nueva_localidad.getClaveLocalidad()),
.....
.....
%>
```