**Bases de Datos Relacionales**

Una **base de datos** o **banco de datos** es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

**Sistema Gestor De Bases de Datos (SGBD)** es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos.

Puesto de forma muy esquemática

El programa típico de Base de Datos permite

Consultas

Altas

Bajas

y Cambios

dicho en ingles

**C**reate

**R**read

**U**pdate

**D**elete

**Modelo relacional**

En este modelo todos los datos son almacenados en relaciones

Una **Base de Datos Relacional**, es una base de datos que cumple con el modelo relacional y esta compuesta por

Tablas (Relaciones en el sentido matemático del termino) y una tabla esta compuesta por

Registros (filas, tuplas) y un registro esta compuesto por

Campos (columnas, atributos)

Dichas tablas se vinculan entre sí por un campo en común

Un campo es la unidad Mínima de Información en la Base de Datos

**Meta Datos**

Son datos que describen otros datos

Ejemplo de datos y meta datos

|  |  |
| --- | --- |
| Datos | Meta Datos |
| Juan Perez | Nombre |
| Av. del Parque 79 | Dirección |
| 5566778899 | Teléfono |
| 33 | Edad |

**Esquema de tabla**

Todo esquema constará de:

* Nombre de la relación (su identificador).
* Nombre de los atributos (o campos) de la relación y sus dominios; el dominio de un atributo o campo define los valores permitidos para el mismo, equivalente al tipo de dato.

Ejemplo Esquema de tabla

Persona(Nombre, Dirección, Teléfono, Edad)

|  |  |
| --- | --- |
| Nombre | varchar(50) |
| Dirección | varchar(50) |
| Teléfono | varchar(50) |
| Edad | entero |

**Diccionario de Datos**

Recoge todos los nombres de los objetos (tablas, vistas, alias...) que contiene una base de datos

**SQL** (Structured Query Language)

Es el lenguaje estandard para Sistemas Gestores De Bases de Datos Relacionales

Ejemplo de una consulta en SQL

Select \* from Persona where edad=33

**JDBC ( JAVA DATABASE CONNECTIVITY )**

**DEFINICIÓN**

JDBC es un API java para ejecutar Frases SQL, basado como OBDC, en X/Open SQL CLI, creado para posibilitar el acceso de los programadores a bases de datos locales y remotas mediante una interfaz común e independiente de la plataforma.

El API JDBC está constituido por el siguiente conjunto de clases e interfaces Java, contenidas en el paquete "java.sql", que forman parte inseparable de la plataforma JDK 1.1.X y que se instalan localmente con la misma:

Tales clases posibilitan conexiones a Bases de Datos, representan frases SQL, definen conjuntos de resultados controlan meta datos, etc. El uso de los métodos de estas clases nos permite, resumiendo, conectarnos con una base de datos enviarle frases SQL y procesar los resultados.

Ejemplo de una aplicación que hace conexión con una base de datos

//ResultApp.java

import java.sql.\*;

import java.util.\*;

class ResultApp {

public static void main (String args [ ]) {

try {

String url = "jdbc:mysql://localhost:3306/", dbName = "test";

Class.forName("com.mysql.jdbc.Driver");

Connection conex=DriverManager.getConnection(url+dbName, "root", "contrasenia" );

Statement stmt=conex.createStatement();

String sql="SELECT \* FROM cursos";

ResultSet result=stmt.executeQuery(sql);

displayResults(result);

conex.close( );

}

catch (Exception ex){

System.out.println(ex);

System.exit(0);

}

}

static void displayResults(ResultSet r) throws SQLException {

ResultSetMetaData rmeta = r.getMetaData();

int numColumns=rmeta.getColumnCount();

for (int i = 1; i <= numColumns; ++i )

System.out.print(rmeta.getColumnName( i )+" ");

while(r.next())

for(int i = 1; i <= numColumns; ++i)

if( i < numColumns)

System.out.print(r.getString( i )+" ");

}//displayResults

}//ResultApp

Obtener una imagen de la base de datos

//mysql --password=gatito --user=root

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import javax.swing.\*;

import java.sql.\*;

import java.util.\*;

//javac -cp .:beanutils.jar:commons-logging.jar ResultApp.java

//java -cp .:mysqlcon.jar:beanutils.jar:commons-logging.jar ResultApp

class ResultApp extends JFrame {

static ImageIcon icon;

static JLabel lPhoto;

static JPanel pnl;

public static void main (String args [ ]) {

String url = "jdbc:mysql://localhost:3306/";

String dbName = "Perracos";

String driverClassName = "com.mysql.jdbc.Driver";

//String url= "jdbc:odbc:Driver={Microsoft Access Driver (\*.mdb)};DBQ=f:\\jdbc\\verduleros.MDB";

ResultApp ra=new ResultApp();

JPanel jp= new JPanel();

jp.setLayout(new FlowLayout());

JScrollPane src\_pane = new JScrollPane (jp);

pnl = (JPanel)ra.getContentPane();

try {

Class.forName("com.mysql.jdbc.Driver");

Connection conex=DriverManager.getConnection(url+dbName, "root", "" );

Statement statement=conex.createStatement();

ResultSet rs=statement.executeQuery("SELECT \* FROM PerroImg");

while(rs.next()) {

System.out.println(rs.getString("nombre"));

byte[] imagedata = rs.getBytes("imagen") ;

Image img = Toolkit.getDefaultToolkit().createImage(imagedata);

icon =new ImageIcon(img);

lPhoto = new JLabel(icon) ;

jp.add(lPhoto);

}

pnl.add(src\_pane);

//conex.close( );

ra.setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);

//ra.pack();

ra.setVisible(true);

}

catch (Exception ex){ System.out.println(ex); System.exit(0); }

}

}//ResultApp