

1.- En el servidor Oracle de cada uno, crear un usuario llamado agenda.

```
CREATE USER agenda IDENTIFIED BY agenda;
GRANT CREATE SESSION, RESOURCE to agenda;
```

Crear una tabla llamada *Contactos*, con los siguientes campos:

<i>idContacto</i>	– NUMBER(4)	(clave principal)
<i>Email</i>	– VARCHAR2(45)	(sin duplicados)
<i>Nombre</i>	– VARCHAR2(45)	(no nulo)
<i>Apellidos</i>	– VARCHAR2(45)	(no nulo)
<i>Telefono</i>	– VARCHAR2(15)	
<i>FechaNacimiento</i>	– DATE	

```
CREATE TABLE contactos(
  idContacto NUMBER(4) PRIMARY KEY,
  email VARCHAR2(45) UNIQUE,
  nombre VARCHAR2(45) NOT NULL,
  apellidos VARCHAR2(45) NOT NULL,
  telefono VARCHAR2(15),
  fechaNacimiento DATE
);
```

(Introducir algunos datos de prueba utilizando SQLDeveloper)

IDCONTACTO	EMAIL	NOMBRE	APELLIDOS	TELEFONO	FECHANACIMIENTO
1	1 pepeGar@gmail.com	José	García Aparicio	985909090	12/03/80
2	2 lolaRius@lalaboral.es	Lola	Rius Pons	666777888	07/12/75
3	3 evaSD@telecable.es	Eva	Santos Díaz	985100200	19/04/00

Crear un pequeño programa Java que saque por pantalla los datos de los contactos introducidos.

```
package tema9;

import java.sql.*;

public class Ejercicio1_1 {
    public static void main(String[] args) throws SQLException{
        Connection conexion = null;
        Statement stmt = null;
        ResultSet resultado = null;
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            String url = "jdbc:oracle:thin:@localhost:1521:xe";
            conexion = DriverManager.getConnection (url, "agenda", "agenda");
            stmt = conexion.createStatement();
            String sql = "SELECT * FROM contactos";
            resultado = stmt.executeQuery(sql);
            System.out.println("IDCONTACTO\tEMAIL\t\tNOMBRE\tAPELLIDOS\tTELEFONO\tFECHA");
            System.out.println("_____");
            while (resultado.next()) {
                int id = resultado.getInt("idContacto");
                String email = resultado.getString("email");
                String nombre = resultado.getString("nombre");
                String apellidos = resultado.getString("apellidos");
                String tel = resultado.getString("telefono");
                Date fecha = resultado.getDate("fechaNacimiento");
                System.out.println(id+"\t"+email+"\t"+nombre+"\t"+apellidos+"\t"+tel+"\t"+fecha);
            }
        } catch (ClassNotFoundException e) {
            System.out.println("Clase no encontrada "+e.getMessage());
        } catch (SQLException e){
            System.out.println("Error SQL "+e.getMessage());
        } finally{
            if (resultado !=null)
                resultado.close();
            if (stmt != null)
                stmt.close();
            if (conexion != null)
                conexion.close();
        }
    }
}
```

Puedo utilizar los números de orden (empezando en 1) en vez de los nombres de los campos (1=idContacto, 2=email, ...)

Console

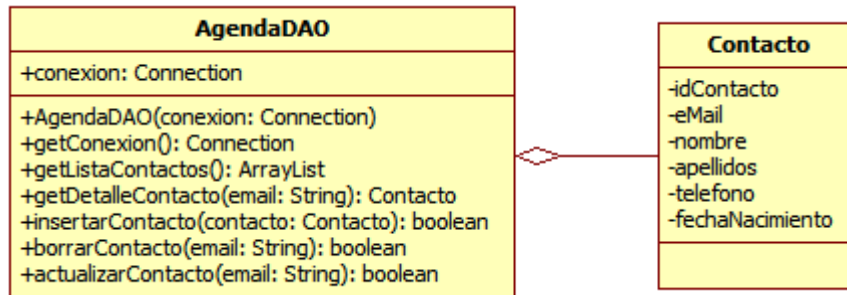
<terminated> Ejercicio1_1 (2) [Java Application] C:\Program Files\Java\jre1.8.0_31\bin\javaw.exe (17/05/2015 21:13:41)

IDCONTACTO	EMAIL	NOMBRE	APELLIDOS	TELEFONO	FECHA
1	pepeGar@gmail.com	José	García Aparicio	985909090	1980-03-12
2	lolaRius@lalaboral.es	Lola	Rius Pons	666777888	1975-12-07
3	evaSD@telecable.es	Eva	Santos Díaz	985100200	2000-04-19

2.- Desarrolla una clase Java que llamaremos **AgendaDAO** que, a partir de la tabla creada proporcione los métodos necesarios para:

- Obtener un ArrayList de objetos Contacto, con los datos almacenados en la tabla Contactos.
- Obtener el detalle de un contacto a partir de su email.
- Insertar un contacto nuevo.
- Borrar un contacto a partir de su email.
- Cambiar los datos de un contacto a partir de su email. Sólo se modificará el teléfono.

El constructor de la clase recibe una conexión ya creada y se la asigna a la propiedad conexión de la clase.



```

/*
 * clase Contacto
 */

package tema9;

public class Contacto {
    private int idContacto;
    private String email;
    private String nombre;
    private String apellidos;
    private String telefono;
    private String fechaNacimiento;

    public String toString() {
        return "[id=" + idContacto + ", email=" + email
            + ", nombre=" + nombre + " " + apellidos
            + ", telefono=" + telefono + ", fechaNacimiento="
            + fechaNacimiento + "]" + "\n";
    }

    public Contacto(int idContacto, String email, String nombre,
        String apellidos, String telefono, String fechaNacimiento) {
        this.idContacto = idContacto;
        this.email = email;
        this.nombre = nombre;
        this.apellidos = apellidos;
        this.telefono = telefono;
        this.fechaNacimiento = fechaNacimiento;
    }

    public int getIdContacto() {
        return idContacto;
    }
}

```

```
public String getEmail() {  
    return email;  
}  
  
public String getNombre() {  
    return nombre;  
}  
  
public String getApellidos() {  
    return apellidos;  
}  
  
public String getTelefono() {  
    return telefono;  
}  
  
public String getFechaNacimiento() {  
    return fechaNacimiento;  
}  
  
}
```

```
/*  
 * clase AgendaDAO  
 */  
  
package tema9;  
  
import java.sql.*;  
import java.util.ArrayList;  
import java.util.Scanner;  
  
public class AgendaDAO {  
    private Connection conexion;  
  
    public AgendaDAO(Connection conexion) {  
        this.conexion = conexion;  
    }  
  
    public Connection getConexion() {  
        return conexion;  
    }  
  
    public boolean insertarContacto(Contacto contacto) throws SQLException {  
        Statement stmt = conexion.createStatement();  
        String sql = "INSERT INTO contactos VALUES("+contacto.getIdContacto()+  
            ", '"+contacto.getEmail()+"', '"+  
            contacto.getNombre()+"', '"+contacto.getApellidos()+"', '"+  
            contacto.getTelefono()+"', '"+  
            contacto.getFechaNacimiento()+"')";  
        stmt.executeUpdate(sql);  
        return true;  
    }  
}
```

```
public ArrayList<Contacto> getListaContactos() throws SQLException{
    ArrayList<Contacto> lista = new ArrayList<>();
    Statement stmt = conexion.createStatement();
    String sql = "SELECT * FROM contactos";
    ResultSet resultado = stmt.executeQuery(sql);
    while (resultado.next()) {
        int id = resultado.getInt("idContacto");
        String email = resultado.getString("email");
        String nombre = resultado.getString("nombre");
        String apellidos = resultado.getString("apellidos");
        String tel = resultado.getString("telefono");
        String fecha = resultado.getDate("fechaNacimiento").toString();
        lista.add(new Contacto(id,email,nombre,apellidos,tel,fecha));
    }
    return lista;
}

public Contacto getDetalleContacto(String email_buscar) {
    Contacto c;
    try{
        Statement stmt = conexion.createStatement();
        String sql = "SELECT * FROM contactos WHERE email='"+email_buscar+"'";
        ResultSet r= stmt.executeQuery(sql);
        r.next();
        c = new Contacto(r.getInt(1),r.getString(2),r.getString(3),
            r.getString(4),r.getString(5),r.getDate(6).toString());
        stmt.close();
        return c;
    }catch (SQLException e) {
        System.out.println("El contacto no está en la agenda ");
        return null;
    }
}

public boolean borrarContacto(String email_borrar) {
    try{
        Statement stmt = conexion.createStatement();
        String sql = "DELETE FROM contactos WHERE email='"+email_borrar+"'";
        if (stmt.executeUpdate(sql)==1)
            return true;
        else
            return false;
    }catch (SQLException e) {
        System.out.println("Error borrando.. "+e.getMessage());
        return false;
    }
}

public boolean actualizarContacto(String email_act) {
    Scanner teclado = new Scanner(System.in);
    boolean actualizo = false;
    try{
        Statement stmt = conexion.createStatement();
        //busco el contacto
        Contacto c = getDetalleContacto(email_act);
        if (c!=null){
            System.out.println("Introduce nuevo teléfono: ");
            String nuevoTfno = teclado.nextLine();
        }
    }
}
```

```

        String sql ="UPDATE contactos SET telefono ='"+nuevoTfno+
            "' WHERE email='"+email_act+"'";
        if (stmt.executeUpdate(sql)==1)
            actualizo = true;
    }
    return actualizo;
} catch (SQLException e) {
    System.out.println("Error actualizando .. "+e.getMessage());
    return false;
}
}
}

```

3.- Crea una clase ejecutable, **PruebaAgenda** que establezca una conexión con el usuario AGENDA de la base de datos y pruebe todos los métodos de la clase AgendaDAO.

```

package tema9;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Scanner;

public class PruebaAgenda {

    public static void main(String[] args) {
        Connection con = null;
        Scanner teclado = new Scanner(System.in);

        try{
            Class.forName("oracle.jdbc.driver.OracleDriver");
            String url = "jdbc:oracle:thin:@localhost:1521:xe";
            con = DriverManager.getConnection (url, "agenda", "agenda");

            AgendaDAO miAgenda = new AgendaDAO(con);

            // Insertar contacto nuevo
            Contacto c1 = new Contacto(4,"manu@gmail.com","Manuel",
                "Perez Souto","900898989","7/5/00");
            if (miAgenda.insertarContacto(c1))
                System.out.println("Contacto insertado correctamente ...");

            //Mostrar Lista de Contactos
            ArrayList<Contacto> listaContactos = miAgenda.getListContactos();
            System.out.println(listaContactos);

            //Buscar contacto por email
            System.out.println("Introduce email a buscar: ");
            String emailBuscar = teclado.nextLine();
            System.out.println("Datos del contacto: ");
            Contacto c2 = null;
            c2 = miAgenda.getDetalleContacto(emailBuscar);
            if (c2!=null)
                System.out.println(c2);
        }
    }
}

```

```
//Borrar contacto dado su email
System.out.println("Introduce email del contacto a borrar: ");
String emailBorrar = teclado.nextLine();
if ( miAgenda.borrarContacto(emailBorrar) )
    System.out.println("Contacto eliminado");
else
    System.out.println("No se ha encontrado");

//Actualizar contacto
System.out.println("Introduce email del contacto a modificar: ");
String emailModificar = teclado.nextLine();
if ( miAgenda.actualizarContacto(emailModificar) )
    System.out.println("Contacto modificado");
else
    System.out.println("No se ha encontrado");

}catch (ClassNotFoundException e) {
    System.out.println("Clase no encontrada "+e.getMessage());
}catch (SQLException e){
    System.out.println("Error SQL "+e.getMessage());
}finally{
    teclado.close();
    if (con != null)
        try {
            con.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
}
```

4.- En tu servidor MySQL crea la tabla de Contactos, estando conectado como el usuario root.

Realiza las modificaciones necesarias para que la aplicación desarrollada en los apartados anteriores funcione con este gestor de bases de datos.

```
CREATE DATABASE IF NOT EXISTS agenda
CHARACTER SET utf8 COLLATE utf8_unicode_ci;

CREATE TABLE contactos(
    idContacto INTEGER PRIMARY KEY,
    email VARCHAR(45) UNIQUE,
    nombre VARCHAR(45) NOT NULL,
    apellidos VARCHAR(45) NOT NULL,
    telefono VARCHAR(15),
    fechaNacimiento DATE
);
```

En el apartado 1, suponiendo que hemos creado la tabla CONTACTOS en la base de datos AGENDA, sustituimos las tres líneas de código en las que especificamos el tipo de conexión:

```
Class.forName("com.mysql.jdbc.Driver");
String url = "jdbc:mysql://localhost:3306/agenda";
conexion = DriverManager.getConnection (url, "root", "");
```

En el apartado 2 no hay que hacer ningún cambio.

En el apartado 3 hacemos lo mismo que en el apartado 1, en el método main.

Además, puesto que en MySQL las fechas se utilizan con el formato yy/mm/dd, también tendremos que modificar el formato de la fecha introducida.

```
Contacto c1 = new Contacto(4,"manu@gmail.com","Manuel",
    "Perez Souto","900898989","00/5/7");
```