

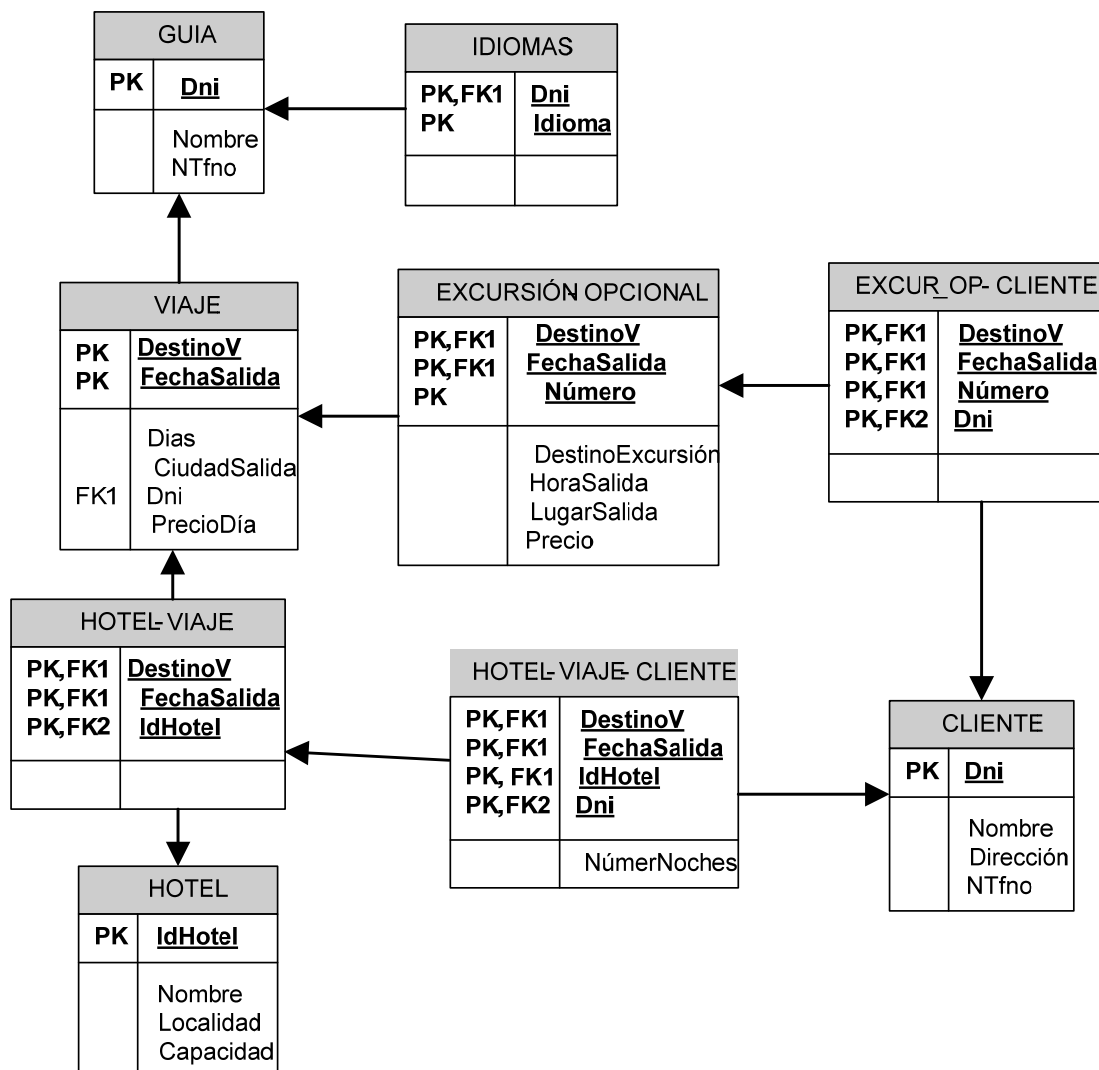
## BASE\_DE\_DATOS - Laboratorio6

### OBJETIVO

- Usando JDBC enviar sentencias SQL a una base de datos relacional, dentro de un programa escrito en Java y practicar los conceptos básicos de transacciones.

### A. BASE\_DE\_DATOS - MySQL

1.Trabajaremos primero con MySql, conectándonos a la base de datos que ya tenemos de Viajes.



### Entorno Eclipse Kleper

Crear un proyecto nuevo con Java utilizando Eclipse.

<File>+<New>+<Java project>. Es necesario para hacer la conexión Java-MySQL, añadir la librería *Mysql-connector-java-5.1.38-bin.jar*. Para ello colocarse encima del proyecto y con el botón derecho: <Properties>+<Java Build Path>+<Libraries>+<Add External Jars> añadir “Mysql-connector-java-5.1.38-bin”.

Crear en la carpeta *src* una nueva clase: <File>+<New>+<Class>+<public void main(String[] args)>, dar un nombre y <Finish>.

Una vez escrito el programa ejecutar: <Run as java application>

Todas las preguntas se harán en el mismo proyecto, pero cada una en clases distintas, <File>+<New>+<Class>...

1. Pregunta 1. Conectar con la base de datos y obtener los datos de los clientes. Base de datos:

```
jdbc:mysql://dif-mysql.ehu.es:3306/BASEDEDATOS
```

El usuario y contraseña serán los del anterior laboratorio.

```
import java.sql.*;
import java.io.*;
import java.util.*;
public class Uno
{public static void main (String[]args){
    System.out.println("Prueba Mysql");
    System.out.println("");
    try{
        Connection Proy_Mysql=
        DriverManager.getConnection
        ("jdbc:mysql:// dif-mysql.ehu.es:3306/DBDCXX",
        "DBDCXX", "DBDCXX");
        Statement s = Proy_Mysql.createStatement();
        ResultSet rs=s.executeQuery

        ESCRIBE EL RESTO DE CÓDIGO

        rs.close();
        Proy_Mysql.close();
    }catch (Exception error)
    {System.out.println(error.getMessage());
    }
}}
```

Pregunta 2. Obtener el nombre del cliente con dni 10000001.

Pregunta 3. Teclear un dni de un cliente y obtener el nombre.

Pregunta 4. Actualizar la base de datos con UPDATE, DELETE y INSERT.

Pregunta 5. En el apartado 4 tratar un error con SQLException

## **B. BASE\_DE\_DATOS - ORACLE**

Trabajaremos con Oracle, conectándonos a una base de datos ya creada: GIPUZKOA. Es necesario para hacer la conexión Java\_Oracle añadir la librería ojdbc6, *<Properties>+<Java Build Path>+<Libraries>+<Add External Jars>* añadir ojdbc6.

GIPUZKOA:

Se ha creado con las siguientes instrucciones:

```
CREATE TABLE producto
```

```
(cod NUMBER(4),
```

```
nombre VARCHAR(15) NOT NULL,
```

```
precio DECIMAL(6,1) NOT NULL,
```

```
cantidad NUMBER(4),
```

```
PRIMARY KEY (cod));
```

```
CREATE TABLE cliente
```

```
(dni NUMBER(8),
```

```
nombre VARCHAR(15) NOT NULL,
```

```
dir VARCHAR(25),
```

```
ciudad VARCHAR(25),
```

```
descuento DECIMAL(3,1) DEFAULT 0.0,
```

```
numPedidos NUMBER(4),
```

```
PRIMARY KEY (dni),
```

```
CONSTRAINT limitePedidos CHECK (numPedidos<6));
```

```
CREATE TABLE pedido
```

```
(dni NUMBER(8),
```

cod NUMBER(4),

fecha DATE,

cantidad NUMBER(4),

Pregunta 6. Conectar con la base de datos GIPUZKOA e imprimir en pantalla los dni de los clientes.

Para oracle:

```
("jdbc:oracle:thin:@vsids11.si.ehu.es:1521:GIPUZKOA", "DBDC_XX",  
"DBDC_XX");
```

### **C. Conexión de las dos bases de datos y copiar de Oracle a MySql**

Pregunta 7. Coger el dni, nombre y direccion de un cliente de la tabla GIPUZKOA y pasarla a la tabla cliente de viajes de Mysql. El nro. de teléfono se queda vacío.

### **D. Transacción en Oracle**

Pregunta 8. Insertar tres tuplas tratándolas dentro de una transacción. Una de ellas será una tupla de un dni ya existente y al tratarlas las tres dentro de una transacción, no se deberán insertar ninguna de las tres.