

## BASES DE DATOS

### LABORATORIO 9

---

<b>Objetivo:</b>	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.
<b>Actividades a realizar:</b>	Realizar consultas SQL desde programas JAVA utilizando JDBC
<b>Actividad a entregar</b>	Entregar por medio de eGela <b>el proyecto Java</b> con cada una de las clases correspondiente a cada ejercicio realizado. Se debe incluir a modo de comentario apellido(s) de quien(es) lo ha(n) realizado.

---

## Base de Datos VIAJES

### VIAJE

<u>Destino</u>	<u>FechaSalida</u>	Dias	PrecioDia	CiudadSalida	DNI
----------------	--------------------	------	-----------	--------------	-----

### HOTEL

<u>IdHotel</u>	Nombre	Localidad	Capacidad
----------------	--------	-----------	-----------

### CLIENTE

<u>Dni</u>	Nombre	Direccion	Ntelefono
------------	--------	-----------	-----------

### GUIA

<u>DNI</u>	Nombre	NTelefono
------------	--------	-----------

### IDIOMA

<u>DNI</u>	Idioma
------------	--------

### EXCURSION\_OPCIONAL

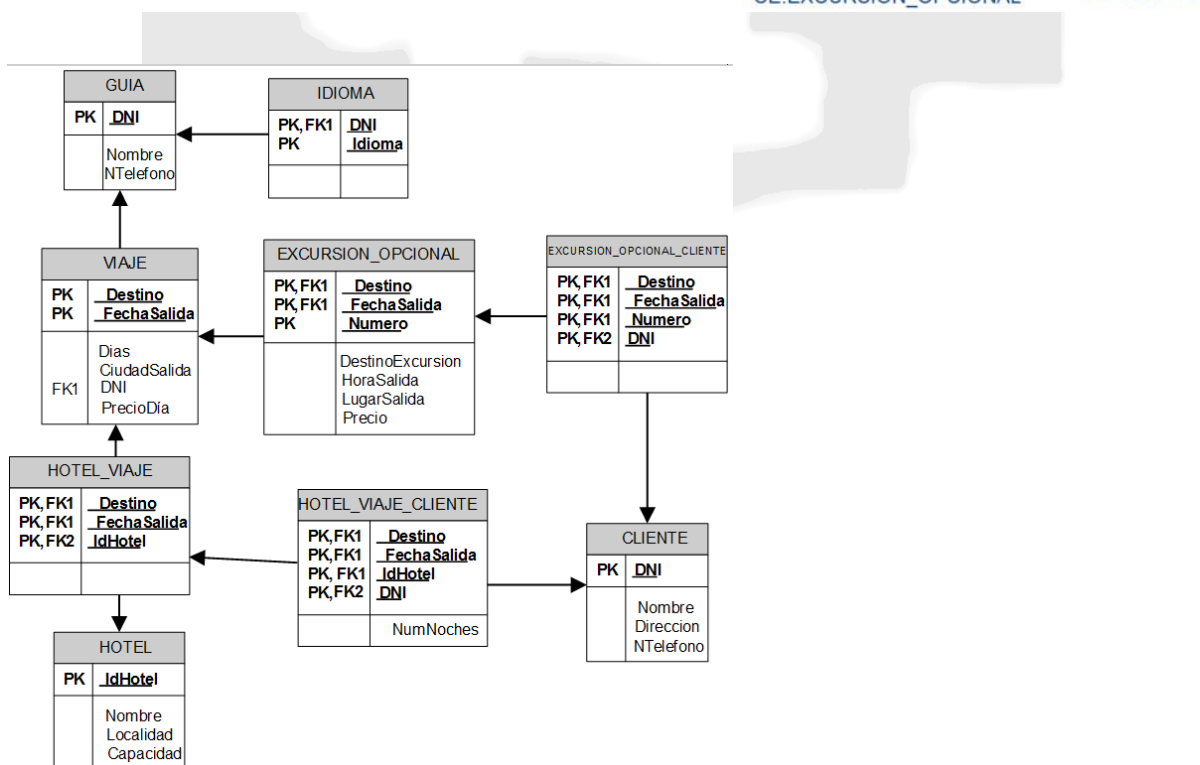
<u>Destino</u>	<u>FechaSalida</u>	<u>Numero</u>	DestinoExcursión	HoraSalida	LugarSalida	Precio
----------------	--------------------	---------------	------------------	------------	-------------	--------

### HOTEL\_VIAJE\_CLIENTE

<u>Destino</u>	<u>FechaSalida</u>	<u>IdHotel</u>	<u>DNI</u>	NumNoches
----------------	--------------------	----------------	------------	-----------

### EXCURSION\_OPCIONAL\_CLIENTE

<u>Destino</u>	<u>FechaSalida</u>	<u>Numero</u>	<u>DNI</u>
----------------	--------------------	---------------	------------



## Entorno Eclipse

Usando el entorno Eclipse, es posible acceder a los contenidos de las tablas de la base de datos de manera interactiva desde el lenguaje de programación Java. Para hacer esto, por supuesto, el entorno debe adaptarse para vincular JAVA y el SGBD.

Crear un proyecto nuevo con Java utilizando Eclipse: *<File>+<New>+<Java project>*. Es necesario para hacer la conexión Java y nuestras bases de datos, añadir las librerías para cada uno de los SGBD con los que queremos trabajar:

- **MySQL:** `Mysql-connector-java-8.0.11.jar`
- **Oracle:** `Ojdbc6.jar`

Para ello colocarse encima del proyecto y con el botón derecho: *<Properties>+<Java Build Path>+<Libraries>+<Add External Jars>* añadirlos.

Para cada ejercicio 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>* o presionar la flecha verde (*Run*).

Haremos todos los ejercicios en el mismo proyecto, pero cada uno en una clase distinta, *<File>+<New>+<Class>...*

## Ejercicio 1.

Conectar con la base de datos Viajes de **MySQL** y de **Oracle** y obtener los datos de los clientes. Para ello utiliza la siguientes URLs y conéctate con el mismo usuario y contraseña que se ha utilizado en laboratorios anteriores, en MySQL DBCXX y en Oracle BDCXX.

`jdbc:mysql://dif-mysql.ehu.es:3306/DBCXX?useSSL=false`

`jdbc:oracle:thin:@vsids11.si.ehu.es:1521:gipuzkoa`

Recuerda los pasos necesarios:

1. Crear conexión (2 conexiones)
2. Crear Objeto Statement (2 objetos)
3. Ejecutar SQL (2 ejecuciones)
4. Recorrer cada uno de los resultados obtenidos e imprimir
5. Liberar recursos y cerrar conexión

**Ejercicio 2.** Conectar con la base de datos Viajes de **MySQL** y obtener datos de clientes utilizando consultas preparadas, `PreparedStatement`.

- Utilizando sentencias preparadas, obtener y escribir los datos del cliente con dni 10000001.
- Dentro del mismo programa, reutilizar la consulta para obtener los datos del cliente 10000004.

Recuerda los pasos necesarios:

1. Crear conexión
2. Preparar consulta
3. Establecer parámetros de consulta
4. Ejecutar SQL
5. Recorrer el resultado e imprimir
6. Reutilización
7. Liberar recursos y cerrar conexión

**Ejercicio 3.** Conectar con la base de datos Viajes de **MySQL** y obtener los datos de un cliente concreto cuyo dni y teléfono se introducen por teclado (*utilizando la clase `Scanner`*). Los pasos son similares a los del ejercicio anterior:

1. Crear conexión
2. Preparar consulta
3. Solicitar parámetros desde el teclado
4. Establecer parámetros de consulta
5. Ejecutar SQL
6. Recorrer el resultado e imprimir
7. Liberar recursos y cerrar conexión

**Ejercicio 4.** Actualizar la base de datos Viajes de **Oracle** con `UPDATE`, `DELETE` e `INSERT`. Escoged qué tupla insertar en qué tabla, qué tupla eliminar y qué tupla actualizar.

**Ejercicio 5.** Añadir el tratamiento de errores con `SQLException` al programa creado en el Ejercicio 4. Por ejemplo, tratar de volver a insertar la tupla recién insertada en el Ejercicio 4. Producirá un error ya que no es posible una nueva tupla con la misma clave primaria. Y este error debe ser tratado como una `SQLException` en este ejercicio. El código de error para la clave repetida en Oracle es ORA-00001 (`getErrorCode () == 1`).

**Ejercicio 6.** EL cliente con DNI 10000001 va a pasar a formar parte de la plantilla de guías de la empresa. En adelante, aparecerá recogido además de como cliente como guía en la BD. Refleja esta operación en la base de datos instalada en el servidor de **Oracle** obteniendo los datos necesarios del servidor de **MySQL**.

Para realizar este ejercicio necesitaréis dos conexiones de bases de datos una sobre el sistema de gestión de bases de datos MySQL, desde donde obtendréis los datos, y otra sobre Oracle, para modificar los datos.

Pasos necesarios:

1. Crear dos conexiones una a cada Base de Datos
2. Crear dos Objetos Statement y PreparedStatement (uno con cada conexión)
3. Ejecutar la consulta de MySQL
4. Recorrer el resultado de 4 y establecer parámetros para la consulta de Oracle
5. Ejecutar la consulta INSERT en Oracle
6. Liberar recursos y cerrar conexión

**Ejercicio 7. Transacciones en ORACLE.** Insertar las tuplas necesarias para repetir los viajes del año 2022 en el año 2023 con las mismas características (mismos hoteles, excursiones y guías). Puedes usar la función `add_months(fecha, 12)`. Ejecútalo dos veces para ver que la segunda ejecución terminaría en rollback.

Pasos necesarios:

1. Crear inicialmente la conexión a *null*
2. Establecer autocommit a *false*
3. Terminar con *commit* en caso de éxito (después de terminar la transacción) o con *rollback* en caso de error (dentro del catch)

**Ejercicio 8. OPCIONAL. Procedimientos almacenados en Oracle y/o MySQL.** Define dos procedimientos almacenados diferente a los ejemplos de las diapositivas y ejecútalos desde Java. En el texto de entrega incluye la definición del procedimiento almacenado.