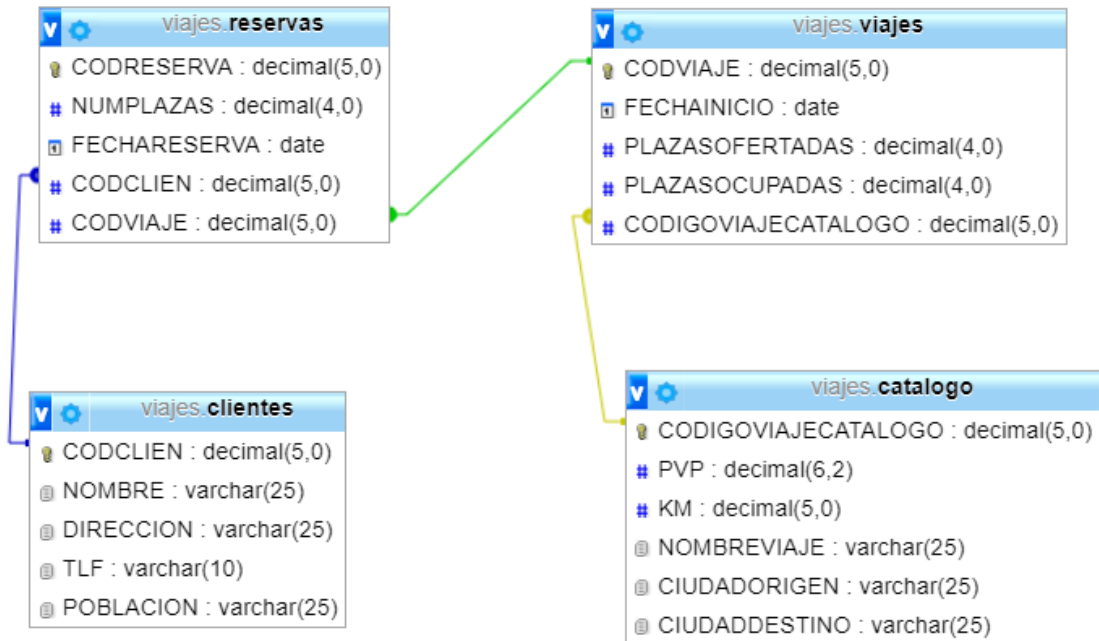


Partimos de la base de datos MySQL **viajes**, usuario y clave **viajes** (debeis crearla y el usuario para acceder a la BD también, se proporciona el script). Para conectarnos se usará la siguiente conexión:

```
Connection conexion = DriverManager.getConnection
    ("jdbc:mysql://localhost/viajes","viajes", "viajes");
```

Las tablas son las siguientes:



**Tabla catálogo.** Contiene la información de los viajes que luego se pueden ofertar. Se identifica por CODIGOVIJECATALOGO, cada uno de estos viajes tiene una ciudad de origen y una ciudad de destino, además del precio del viaje y los kilómetros del viaje.

**Tabla viajes.** Contiene la información de los viajes que se ofertan. Se identifica por CODVIAJE, la FECHAINICIO nos dice cuando se realiza el viaje. PLAZASOFERTADAS son las plazas que salen para el viaje, PLAZASOCUPADAS las ocupadas hasta ese momento.

**Tabla reservas.** Contiene la información de las reservas que realizan los clientes. El cliente puede reservas varias plazas en NUMPLAZAS. También contiene el día que hace la reserva FECHARESERVA.

**Tabla clientes.** Contiene la información de los clientes que realizan las reservas. Se identifica por CODCLIEN.

### REALIZA UNA CLASE JAVA QUE CUMPLA LA SIGUIENTE FUNCIONALIDAD (CREA UN PROYECTO ECLIPSE NUEVO PARA HACER EL EJERCICIO).

- Desde la línea de argumentos de **main()** se recibe un nombre de ciudad. Si no se reciben argumentos o el número de argumentos es incorrecto se mostrará mensaje y finalizará el programa.
- Con la ciudad recibida se debe mostrar por pantalla la siguiente información donde se visualicen todos los viajes con origen esa ciudad y el detalle de las reservas de esos viajes. El listado debe estar ordenado por código de viaje, y dentro de cada viaje por código de cliente.

Si la ciudad origen no existe no se realizará el listado, pero el resto del proceso sí se realizará.

Si en el viaje no hay clientes se debe visualizar mensaje indicándolo, no mostrar el resto de cabeceras y datos del listado.

## RA2 - VIAJES:

El listado para una ciudad origen que tiene viajes es el siguiente (los campos en negrita marcados con (\*) y (\*\*) se calculan como se indica después), el IMPORTE es = NUMPLAZAS \* PVP:

Nº de viajes con ciudad origen xxxxxxxxx: **xxx (\*)**

```
=====
COD-VIAJE: XXXXX          NOMBRE VIAJE: XXXXXXXXXXXXXXXXXXXXXXXXXXXX
PVP VIAJE: XXX.XXX,XX    FECHA INICIO: (DD-MM-YYYY)  PLAZAS OFERTADAS: XXX
CIUDAD DESTINO: MUNICH
=====
```

COD-CLI	NOMBRE	NUM-PLAZAS	IMPORTE
xxxxx	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	xxx	xxx.xxx,xx
xxxxx	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	xxx	xxx.xxx,xx
xxxxx	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	xxx	xxx.xxx,xx

TOTALES POR VIAJE: **xxx xxx.xxx,xx (\*\*)**

Cliente/s con más importe: xx

```
=====
COD-VIAJE: XXXXX          NOMBRE VIAJE: XXXXXXXXXXXXXXXXXXXXXXXXXXXX
PVP VIAJE: XXX.XXX,XX    FECHA INICIO: (DD-MM-YYYY)  PLAZAS OFERTADAS: XXX
CIUDAD DESTINO: MUNICH
=====
```

COD-CLI	NOMBRE	NUM-PLAZAS	IMPORTE
xxxxx	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	xxx	xxx.xxx,xx
xxxxx	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	xxx	xxx.xxx,xx
xxxxx	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	xxx	xxx.xxx,xx

TOTALES POR VIAJE: **xxx xxx.xxx,xx (\*\*)**

Cliente/s con más importe: xx

```
=====
COD-VIAJE: XXXXX          NOMBRE VIAJE: XXXXXXXXXXXXXXXXXXXXXXXXXXXX
PVP VIAJE: XXX.XXX,XX    FECHA INICIO: (DD-MM-YYYY)  PLAZAS OFERTADAS: XXX
CIUDAD DESTINO: MUNICH
=====
```

ESTE CODIGO DE VIAJE NO TIENE CLIENTES....

```
=====
. . . . .
. . . . .
=====
```

CREAR UNA FUNCIÓN Y UN PROCEDIMIENTO ALMACENADO EN LA BASE DE DATOS PARA MOSTRAR LOS DATOS EN NEGRITA DEL LISTADO (\*) Y (\*\*)

Nº de viajes con ciudad origen xxxxxxxxx: **xxx (\*)**

EL VALOR MARCADO EN NEGRITA SE OBTIENE DE UNA **FUNCIÓN ALMACENADA** QUE TENDRÁS QUE CREAR EN LA PROPIA CLASE JAVA. DEBES AVERIGUAR QUE PARÁMETROS SE NECESITAN PARA OBTENER EL CÁLCULO REQUERIDO.

TOTALES POR VIAJE: **xxx xxx.xxx,xx (\*\*)**

LOS VALORES EN NEGRITA TOTAL NÚMERO DE PLAZAS Y TOTAL IMPORTE DEL LISTADO SE OBTIENEN DE UN **PROCEDIMIENTO ALMACENADO** QUE TENDRÁS QUE CREAR EN LA PROPIA CLASE. TENEIS QUE AVERIGUAR LOS PARÁMETROS NECESARIOS PARA OBTENER LOS CÁLCULOS REQUERIDOS.

## RA2 - VIAJES:

- Además tendrás que actualizar las PLAZASOCUPADAS de la tabla VIAJES. Para ello tendremos en cuenta la tabla RESERVAS, que es donde tenemos el número de plazas reservadas de cada reserva-viaje. Las PLAZASOCUPADAS de un viaje, actualizadas serán la suma de lo que había en la columna + las plazas que han reservado los clientes en ese viaje.

Si las PLAZASOCUPADAS totales superan a las PLAZASOFERTADAS del viaje se grabará un registro en la tabla INCIDENCIAS (que tendrás que crear en la propia clase Java, ojo porque puede existir...) y no se actualizarán las PLAZASOCUPADAS. Las columnas para esta tabla de incidencias deben tener la siguiente información: el número de incidencia (que debe empezar en 1, debes ir sumando 1 para cada fila, y no se debe repetir), el código de viaje que provoca la incidencia, comentarios, en esta columna debéis poner "*PLAZASOCUPADAS = valor, superan a las PLAZASOFERTADAS = valor*"; en *valor* indicar el número de plazas de cada caso. Y la columna fecha en la que almacenaremos la fecha actual.

Visualizar mensajes en pantalla de lo que vaya ocurriendo, por ejemplo, Viaje actualizado, Incidencia registrada, de esta manera:

PROCESO ACTUALIZACION:

```
VIAJE xx, ACTUALIZADO.  
VIAJE xx, ACTUALIZADO.  
INCIDENCIA xx, REGISTRADA.  
INCIDENCIA xx, REGISTRADA.  
. . . . .
```