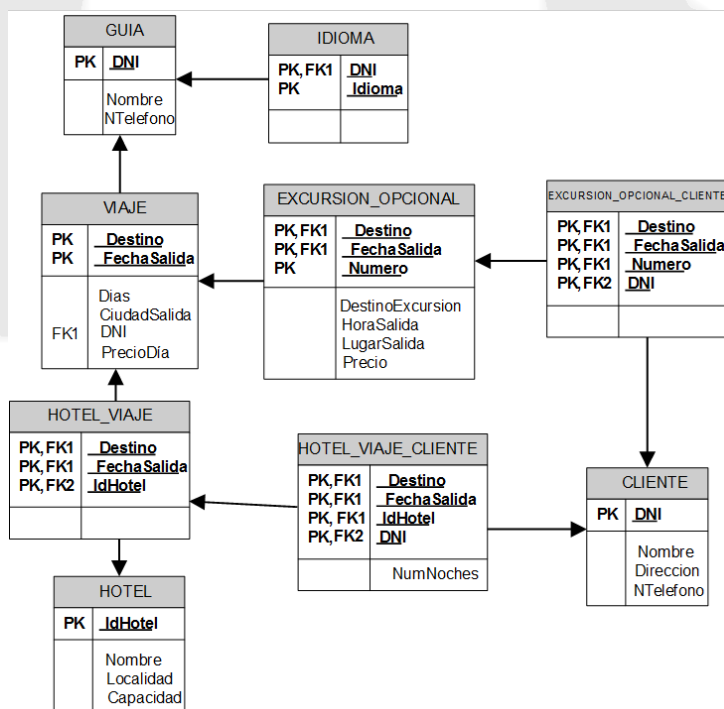
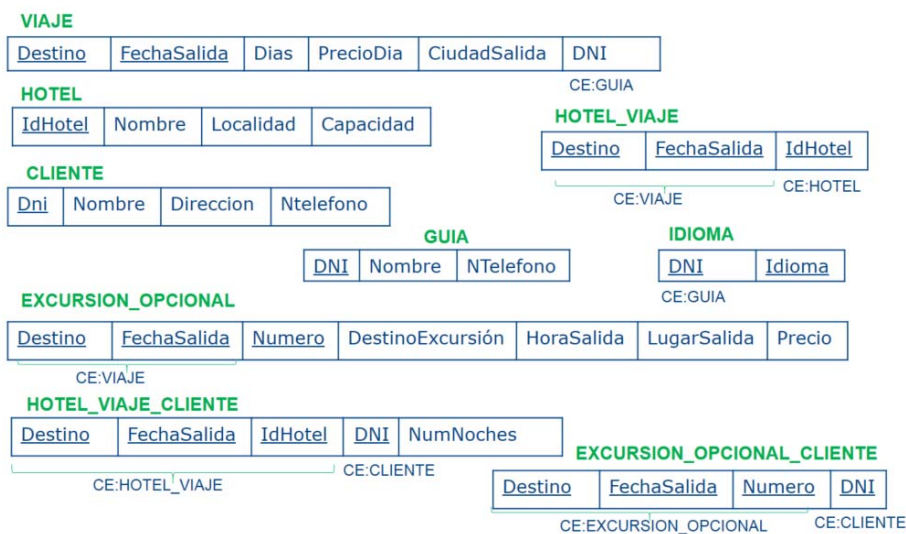




## BASES DE DATOS

### LABORATORIO 4

<b>Objetivo:</b>	Utilizar SQL para realizar consultas en una BD relacional.
<b>Actividades a realizar:</b>	<ol style="list-style-type: none"><li>1.1. Importar Bases de Datos</li><li>1.2. Creación de tablas y definición de claves primarias.</li><li>1.3. Establecimiento de reglas de integridad referencial</li><li>1.4. Realizar consultas SQL</li></ol>
<b>Actividad a entregar</b>	<p>Entregar por medio de eGela <b>un documento</b> pdf con las consultas realizadas y sus resultados. Escribe las consultas separando las cláusulas en diferentes líneas e incluyendo las condiciones de concatenación de tablas en la cláusula FROM.</p> <p>El documento ha de incluir tanto en el nombre como dentro del documento la autoría.</p>





## Tarea 0. Conexión al servidor.

Instalado en la Facultad. Para empezar a trabajar es necesario conectarse al servidor. Para ello conéctate al servidor <http://dif-mysql.ehu.es/>. A continuación, deberás introducir el nombre de usuario y lo mismo como contraseña.

## Tarea 1. Creación de una BD.

**Aviso:** En este laboratorio ya está creado un esquema de base de datos. Es una base de datos vacía cuyo nombre coincide con vuestro nombre de usuario (DBC01, DBC02, ...). Sólo necesitas hacer clic en dicho esquema para crear las tablas y trabajar con esa base de datos.

Si no fuera así deberías crear una BD. Para crear una base de datos introduciremos un nombre en la casilla “Crear una nueva base de datos”. No será necesario especificar nada en “Cotejamiento”; trabajaremos con el que el sistema elije por defecto “utf8\_general\_ci”.

## Tarea 2. Importar Base de Datos

En eGela dispones de la estructura SQL de la base de datos. Importa esa base de datos de MySQL. Para hacer esto utilizar la opción *IMPORTAR* con dicho archivo y hacer clic en *CONTINUAR*.

## Tarea 3. Creación de tablas y definición de claves primarias.

Olvidamos definir la tabla *Hotel\_Viaje\_Cliente* en la BD recién importada. Crea esta tabla usando el entorno gráfico MySQL (Estructura, Crear Tabla). A cada atributo asígnale el tipo de datos más adecuado entre los disponibles y coherentes con los tipos de las otras tablas. Algunos tipos de datos implican especificaciones adicionales. Por ejemplo, si se indica que un atributo es de tipo *VARCHAR*, necesariamente deberemos especificar la longitud máxima en número de caracteres.

Una vez definidos todos los atributos de una tabla tienes que definir su clave primaria. Cuidado con las claves primarias formadas por más de un atributo. Mira a ver cómo puedes definir la clave primaria, la característica de unicidad, etc. Para que el comportamiento de las claves extranjeras sea el adecuado es necesario crear las tablas con el **motor InnoDB**, será el que pone por defecto.

## Tarea 4. Establecimiento de reglas de integridad (RI) referencial

Define las restricciones de integridad referencial necesarias para la tabla *Hotel\_Viaje\_Cliente*. Para definir restricciones de integridad referencial entre tablas, es necesario seleccionar aquella tabla que contenga clave/s extranjera/s y pulsar la opción *VISTA DE RELACIONES*. Se relacionará cada atributo que sea clave extranjera con el atributo correspondiente de la tabla en la que aparezca como clave primaria y se definirá el comportamiento tanto en modificación como en borrado. En nuestro caso, y por definición arbitraria, definiremos borrado y modificación en cascada.



## Después insertar las siguientes tuplas en la tabla:

```
INSERT INTO `hotel_viaje_cliente` (`Destino`, `FechaSalida`, `IdHotel`, `DNI`,  
`NumNoches`) VALUES  
( 'Asilah', '2021-05-03', 'h20', '10000015', 7),  
( 'Auckland', '2021-04-01', 'h03', '10000005', 4),  
( 'Auckland', '2021-04-01', 'h03', '10000008', 4),  
( 'Auckland', '2021-04-01', 'h03', '10000023', 10),  
( 'Barcelona', '2021-05-04', 'h21', '10000004', 3),  
( 'Donostia', '2019-01-01', 'h02', '10000018', 4),  
( 'Donostia', '2019-01-01', 'h02', '10000004', 4),  
( 'Donostia', '2022-01-02', 'h02', '10000013', 1),  
( 'Everest', '2021-11-11', 'h08', '10000001', 12),  
( 'Haro', '2017-06-02', 'h07', '10000020', 1),  
( 'India', '2024-12-01', 'h18', '10000001', 4),  
( 'India', '2024-12-01', 'h18', '10000003', 4),  
( 'Mordor', '2020-04-11', 'h16', '10000008', 5),  
( 'Mordor', '2020-04-11', 'h16', '10000026', 5),  
( 'Namibia', '2020-11-21', 'h09', '10000012', 4),  
( 'Namibia', '2020-11-21', 'h09', '10000013', 4),  
( 'Namibia', '2020-11-21', 'h09', '10000016', 4),  
( 'Namibia', '2020-11-21', 'h09', '10000018', 4),  
( 'Namibia', '2020-11-21', 'h09', '10000019', 4),  
( 'Namibia', '2020-11-21', 'h09', '72515655', 4),  
( 'Namibia', '2020-11-21', 'h09', '72515657', 5),  
( 'Pekin', '2021-05-02', 'h05', '10000001', 20),  
( 'Pekin', '2021-05-02', 'h05', '10000023', 20),  
( 'Pekin', '2022-01-11', 'h01', '10000009', 1),  
( 'Peru', '2021-06-01', 'h04', '10000009', 10),  
( 'Piondjang', '2019-01-06', 'h13', '10000024', 4),  
( 'Piondjang', '2019-01-06', 'h13', '10000012', 4),  
( 'Rias Baixas', '2021-05-19', 'id107', '10000020', 4),  
( 'Rias Baixas', '2021-05-19', 'id108', '10000019', 7),  
( 'Rias Baixas', '2022-05-19', 'id107', '10000020', 4),  
( 'Rotorua', '2020-03-01', 'id101', '10000002', 1),  
( 'Sahara', '2020-09-10', 'h10', '10000008', 1),  
( 'Sahara', '2020-09-10', 'h10', '10000017', 1),  
( 'Sahara', '2020-09-10', 'h10', '10000021', 3),  
( 'Sahara', '2020-11-02', 'h10', '10000011', 2),  
( 'Sahara', '2020-11-02', 'h10', '10000021', 2),  
( 'Senegal', '2019-03-01', 'h06', '10000004', 1),  
( 'Senegal', '2022-04-01', 'h17', '10000010', 3),  
( 'Senegal', '2022-04-01', 'h17', '10000014', 3),  
( 'Toledo', '2020-05-01', 'h19', '10000001', 1),  
( 'Toledo', '2020-05-01', 'h19', '10000002', 1),  
( 'Toledo', '2020-05-01', 'h19', '10000026', 1),  
( 'Toledo', '2020-05-01', 'id111', '10000026', 1),  
( 'Toledo', '2021-05-01', 'id111', '10000002', 1),  
( 'Toledo', '2021-05-01', 'id111', '10000006', 1),  
( 'Toledo', '2021-05-01', 'id111', '10000026', 1),  
( 'Torres del Paine', '2022-11-11', 'h14', '10000022', 1);
```

## Tarea 5. Consultas SQL en MySQL.

Se debe incluir en el documento de entrega tanto la consulta como el resultado de la consulta.

- 5.1. Obtener cuántos guías diferentes han trabajado como guías en los viajes desde Mordor.

Resultado: 6

- 5.2. Por cada cliente que ha viajado a Namibia y ha contratado excursiones obtener cuántas excursiones ha hecho y cuánto se ha gastado en total en ellas.

Resultado:

10000012	Inma Arruti	1	60
10000018	Odin Luengo	1	60
10000019	Bixente Astiz	1	60
72515655	Mikel Arteta	3	190
72515657	Patxi Valverde	2	100
10000013	Xabier Luke	1	40
10000016	Ane Garmendia	1	40

- 5.3. Obtener el dni y nombre de guías que hablan más de 3 idiomas. En el resultado aparecerá también el número de idiomas que hablan y se ordenará descendientemente por esta columna.

Resultado:

72515673	Bienvenido Artetxe	10
72515650	Ainhoa Zapiain	5
72515667	Igor Zalakain	4

- 5.4. Obtener los viajes que tiene más de un hotel asignado y tales que todos ellos están en la misma localidad. En el resultado aparecerá el viaje y el número de hoteles asignados

Resultado:

Donostia	2019-01-01	2
Rias Baixas	2021-05-19	2
Rias Baixas	2022-05-19	2
Rotorua	2020-03-01	2
Rotorua	2020-04-01	2
Toledo	2020-05-01	2
Toledo	2021-05-01	2
Vigo	2021-03-19	2
Vigo	2021-04-19	2

- 5.5. a) Por cada viaje en el que haya participado algún/a cliente acompañado por un/a guía que tenga su mismo nombre y apellido, obtener el identificador del viaje junto con el dni y nombre del cliente.

Resultado:

Rotorua	2020-03-01	10000002	Mikel Landa
---------	------------	----------	-------------

- 5.5. b) Por cada viaje en el que haya participado algún/a cliente acompañado por un/a guía que tenga su mismo nombre de pila, obtener el identificador del viaje junto con el dni y nombre del cliente. Puedes utilizar, por ejemplo, la función SUBSTRING\_INDEX (<https://dev.mysql.com/doc/refman/5.7/en/string-functions.html>)

Resultado:

Rotorua	2020-03-01	10000002	Mikel
Mordor	2020-04-11	10000026	Bienvenido