

Bases de datos

Ejercicios UD 03 - Normalización - DDL y DCL

Francisco Jesús Delgado Almirón



Ciclo formativo de grado superior
Desarrollo de
aplicaciones web

Pautas para la entrega de la relación de ejercicios:

- Esta relación de ejercicios se debe entregar obligatoriamente en formato sql.
- Cada relación de ejercicios deberá incluir los siguientes datos: el nombre del alumno, el Ciclo Formativo, el Módulo Profesional y el curso académico. Por ejemplo, en un fichero de texto.

Por ejemplo:

- **Alumno:** Tu nombre y apellidos
- **C.F.G.S. Desarrollo de Aplicaciones Web**
- **Módulo Profesional:** Bases de datos
- **Curso académico:** Curso académico en el que nos encontremos
- El fichero de los ejercicios deberá subirse a la plataforma Moodle con el siguiente formato de nombre: MóduloFormativo_NombreAlumno_UDX.extensión, por ejemplo:

BADAT_FranciscoJesusDelgadoAlmiron_UD03.sql

En el caso de que no se sigan las pautas de entrega anteriores se penalizará de la forma que se crea conveniente al alumno.

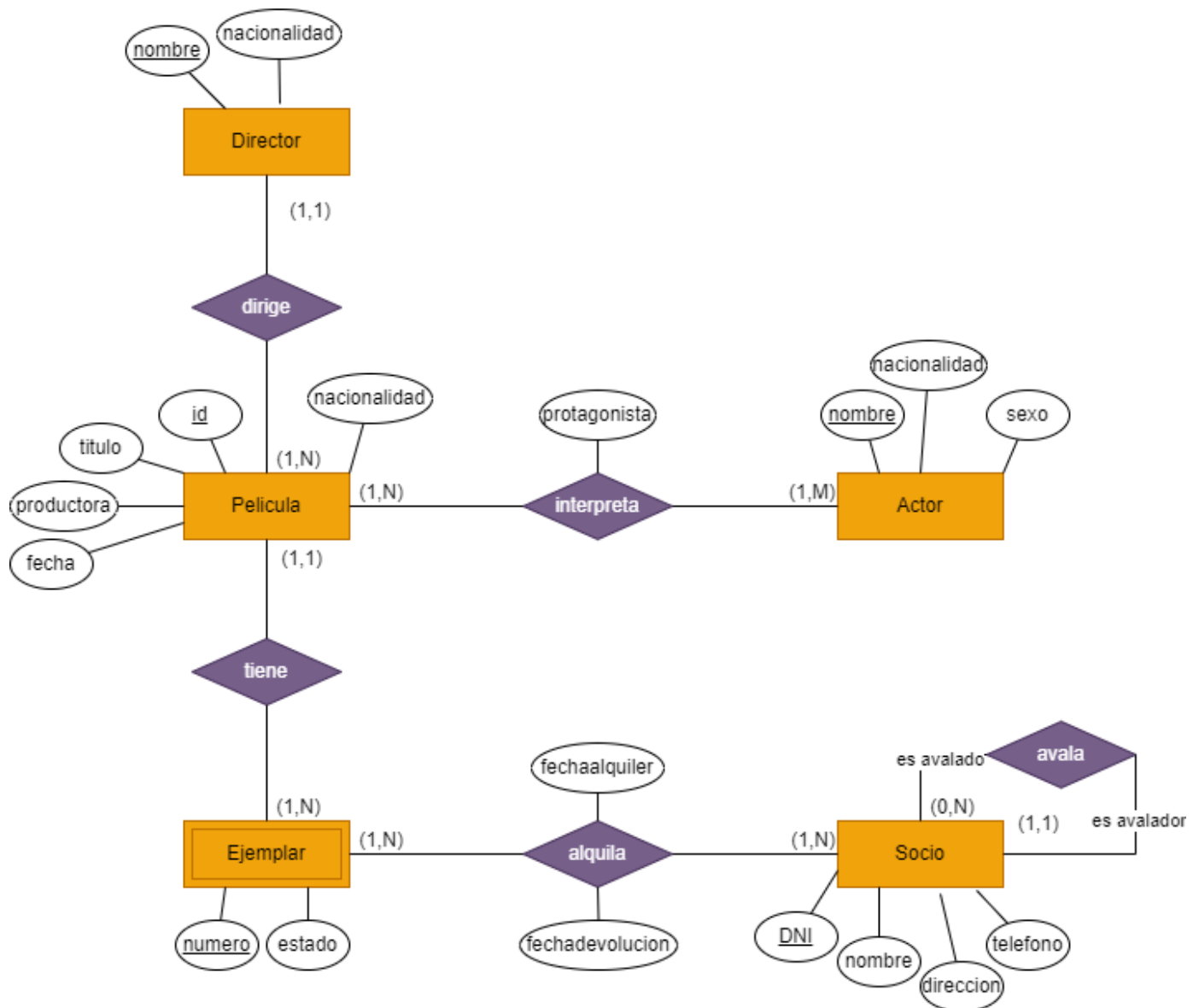
Esta relación de ejercicios contempla los siguientes Resultados de Aprendizaje y Criterios de Evaluación:

RA 2. Crea bases de datos definiendo su estructura y las características de sus elementos según el modelo relacional. [abcdefgh]

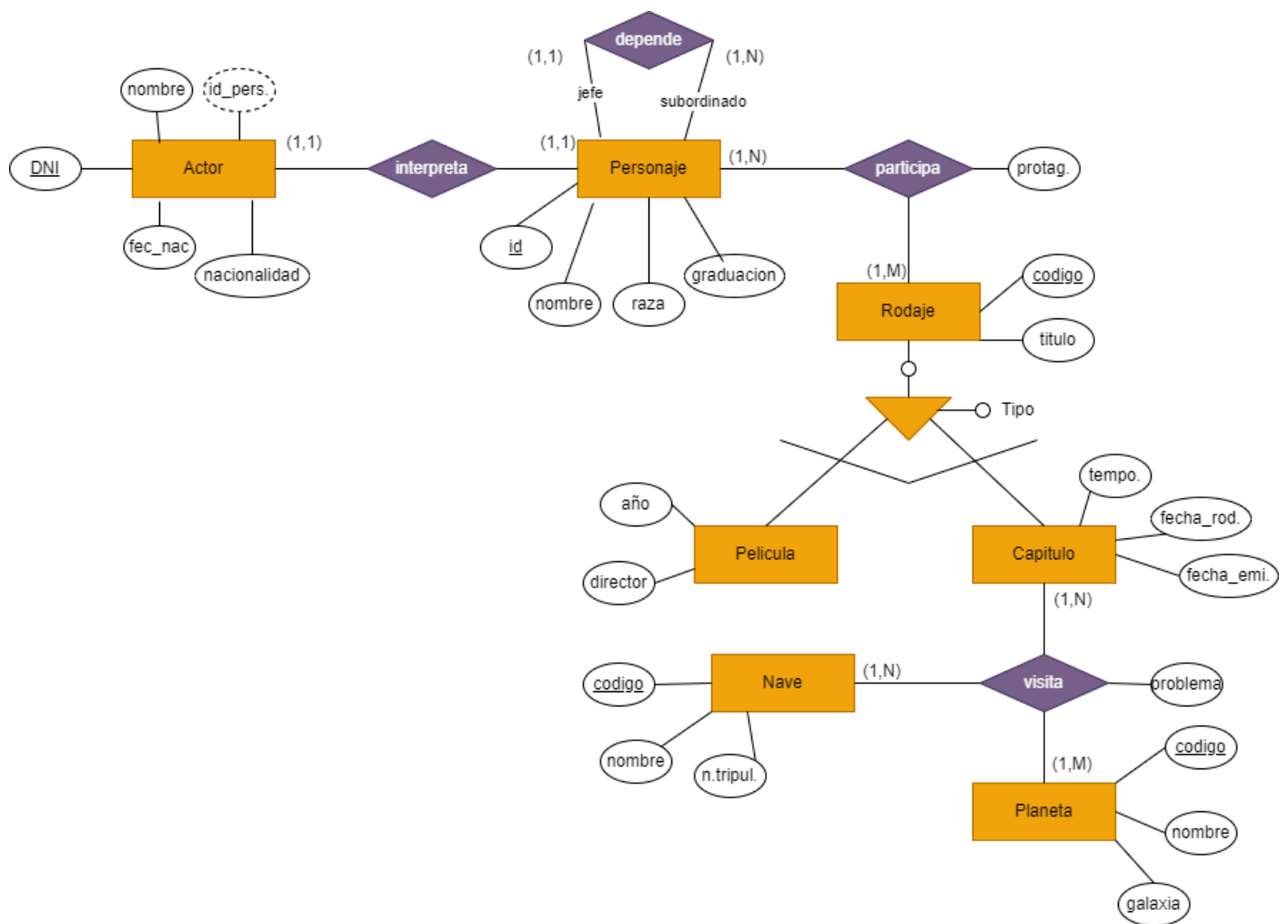
RA 6. Diseña modelos relacionales normalizados interpretando diagramas entidad/relación. [abcdefgh]

EJERCICIOS DE NORMALIZACIÓN Y PASO A TABLAS

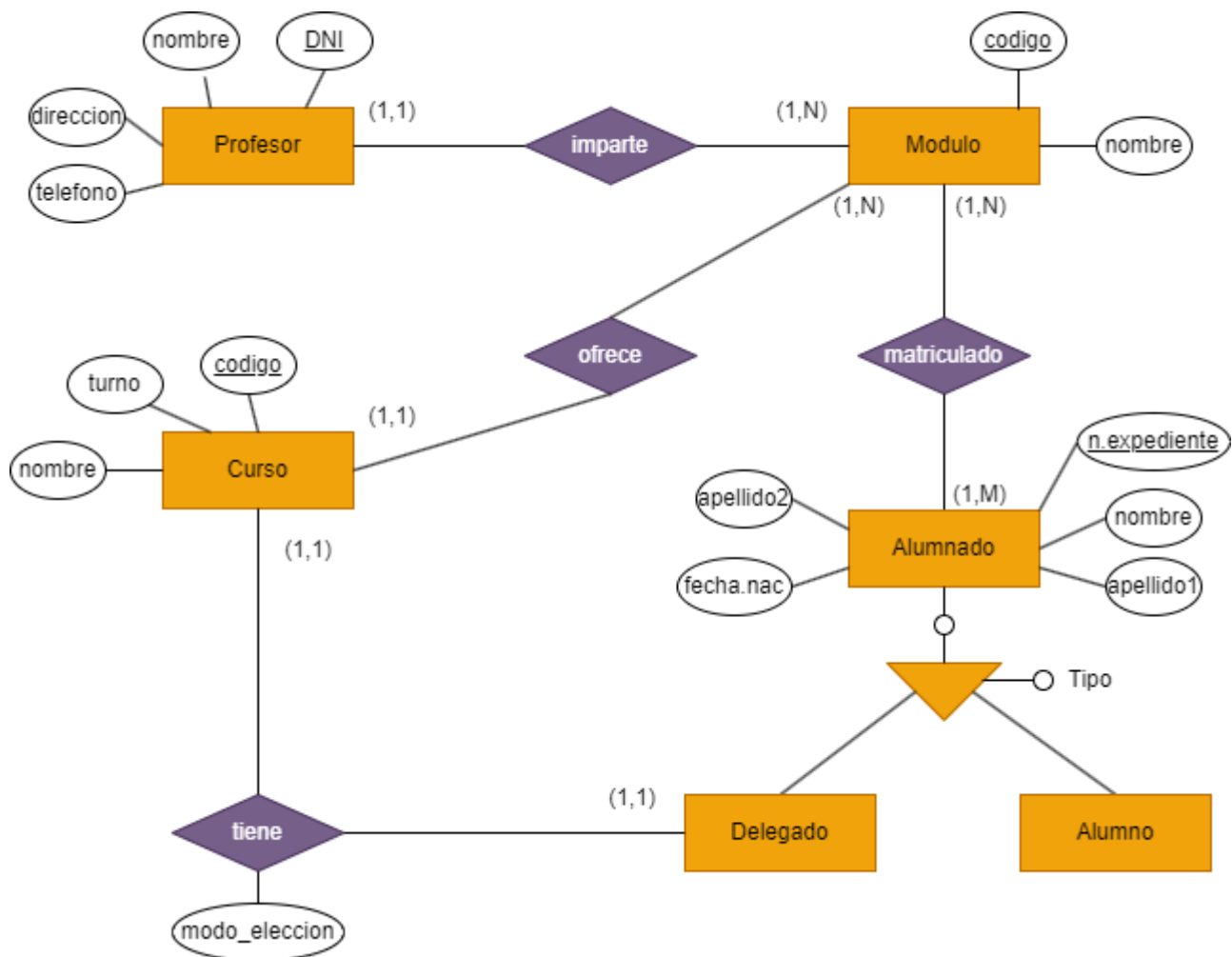
Ejercicio 01: Obtén las tablas correspondientes al siguiente Modelo Entidad/Relación mediante el método de normalización.



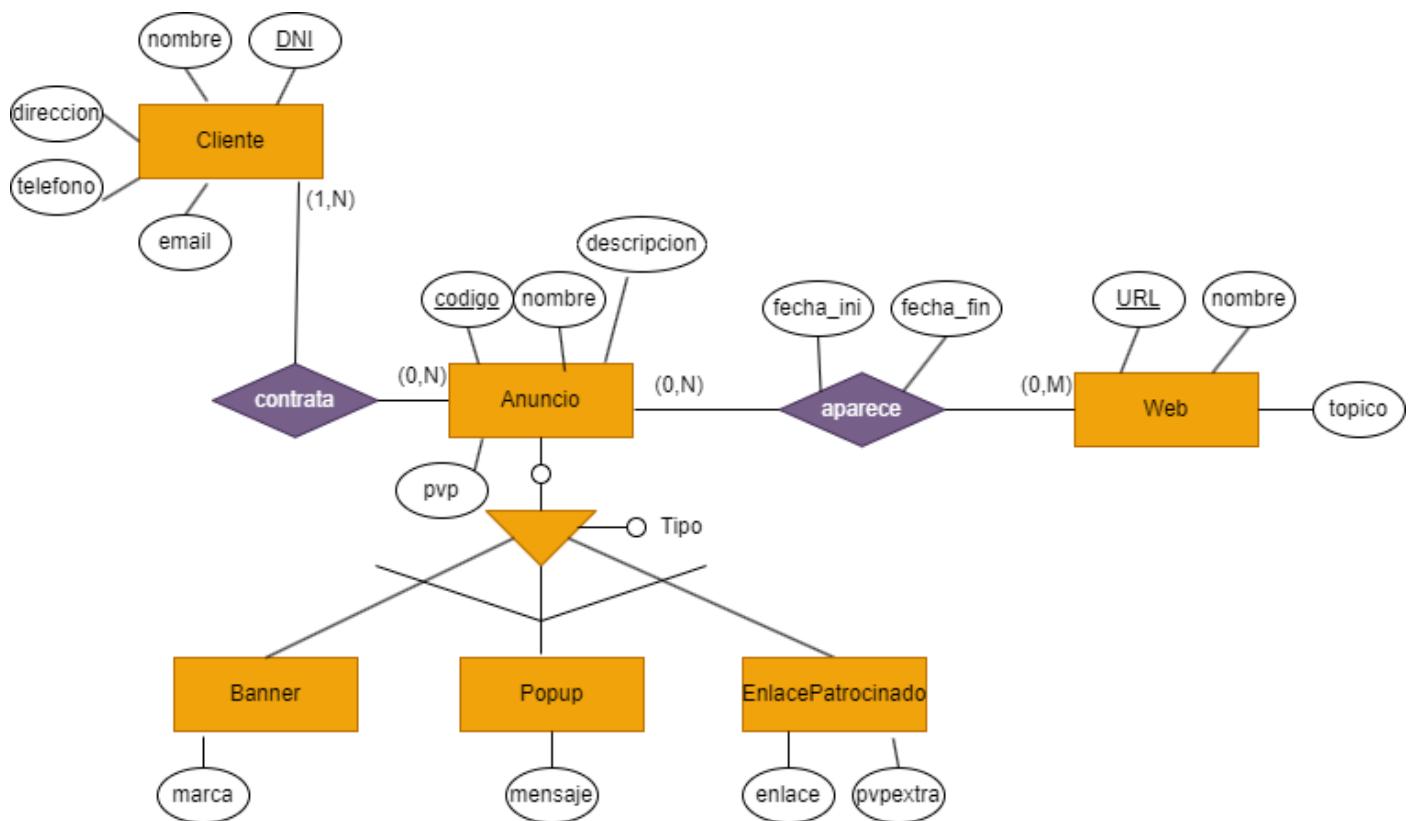
Ejercicio 02: Obtén las tablas correspondientes al siguiente Modelo Entidad/Relación mediante el método de normalización.



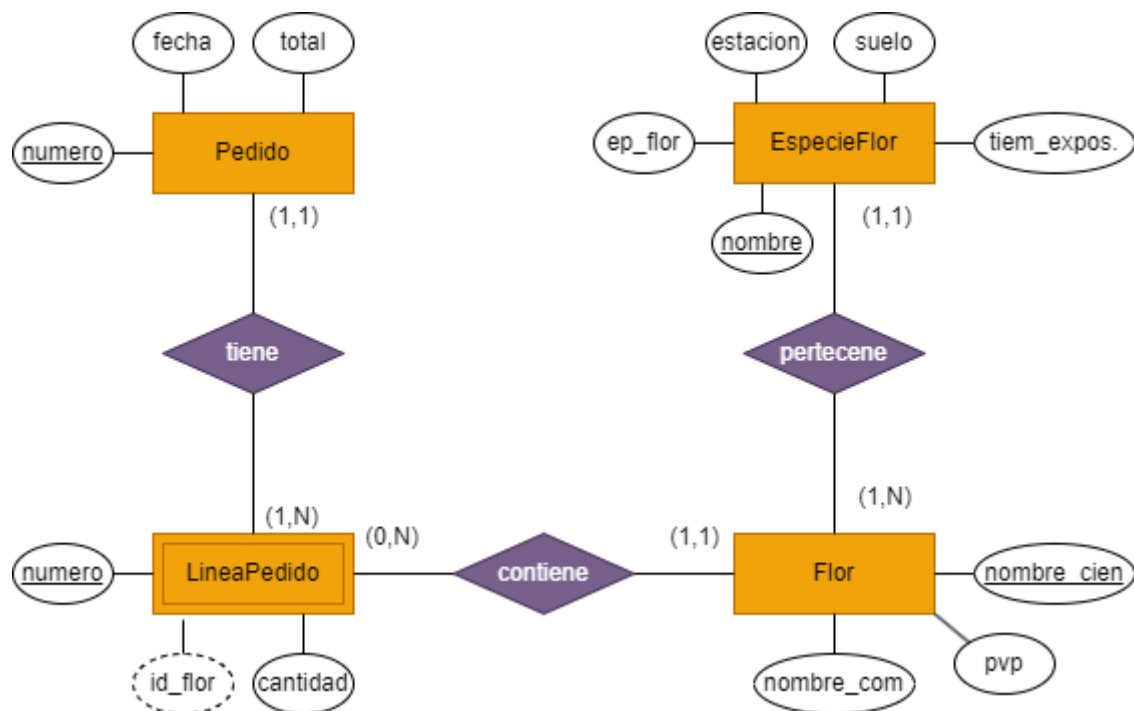
Ejercicio 03: Obtén las tablas correspondientes al siguiente Modelo Entidad/Relación mediante el método de normalización.



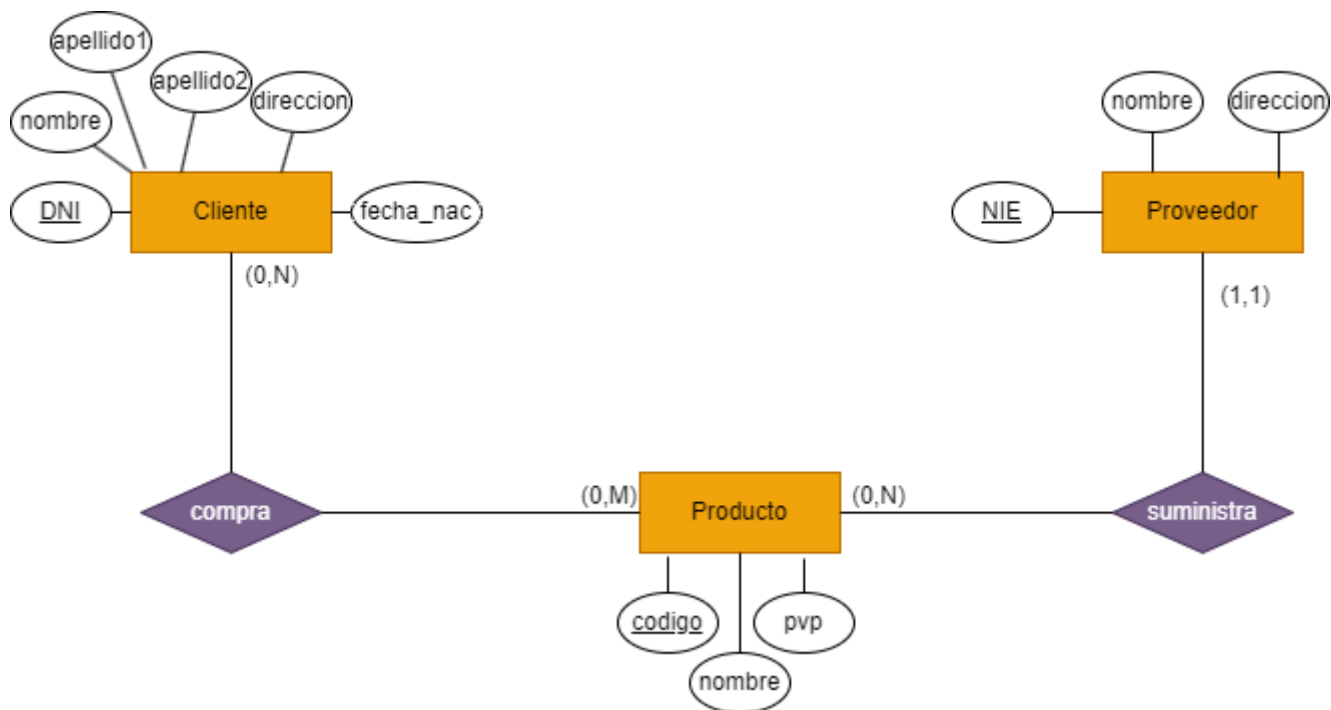
Ejercicio 04: Obtén las tablas correspondientes al siguiente Modelo Entidad/Relación mediante el método de normalización.



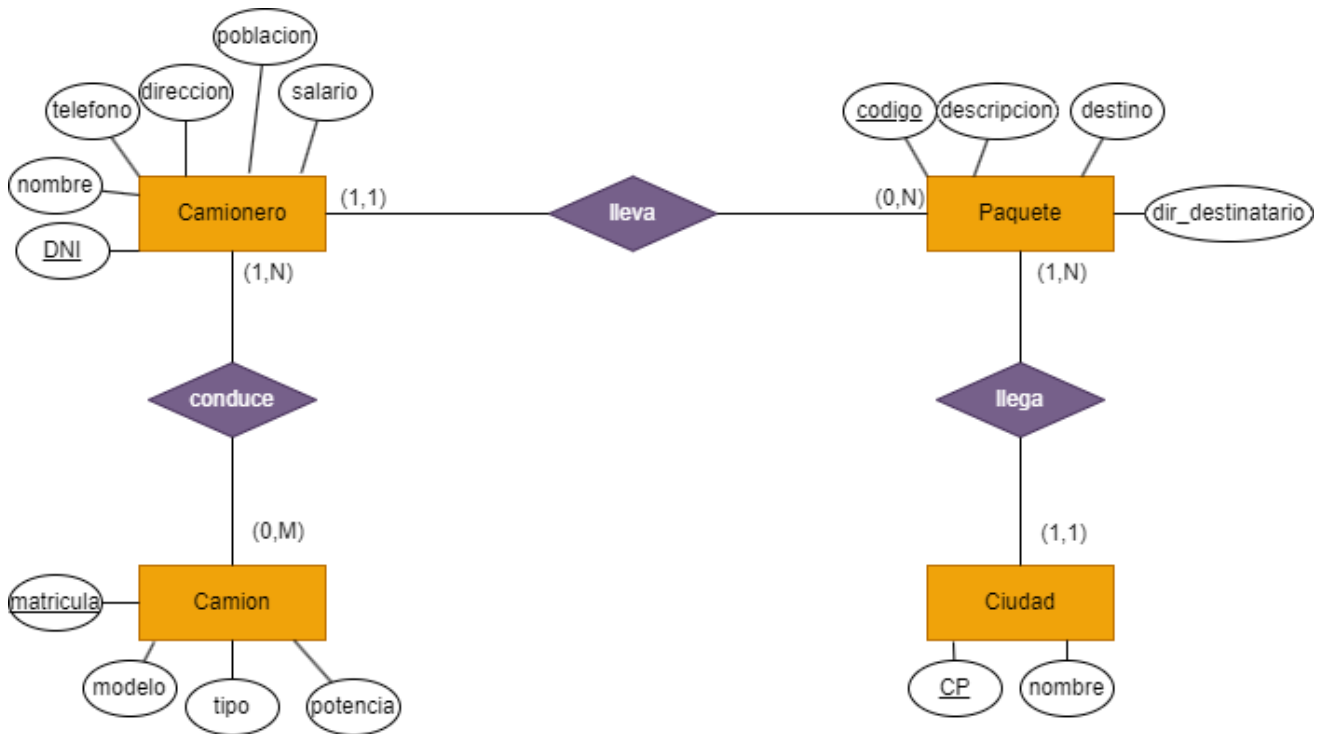
Ejercicio 05: Obtén las tablas correspondientes al siguiente Modelo Entidad/Relación mediante el método de normalización.



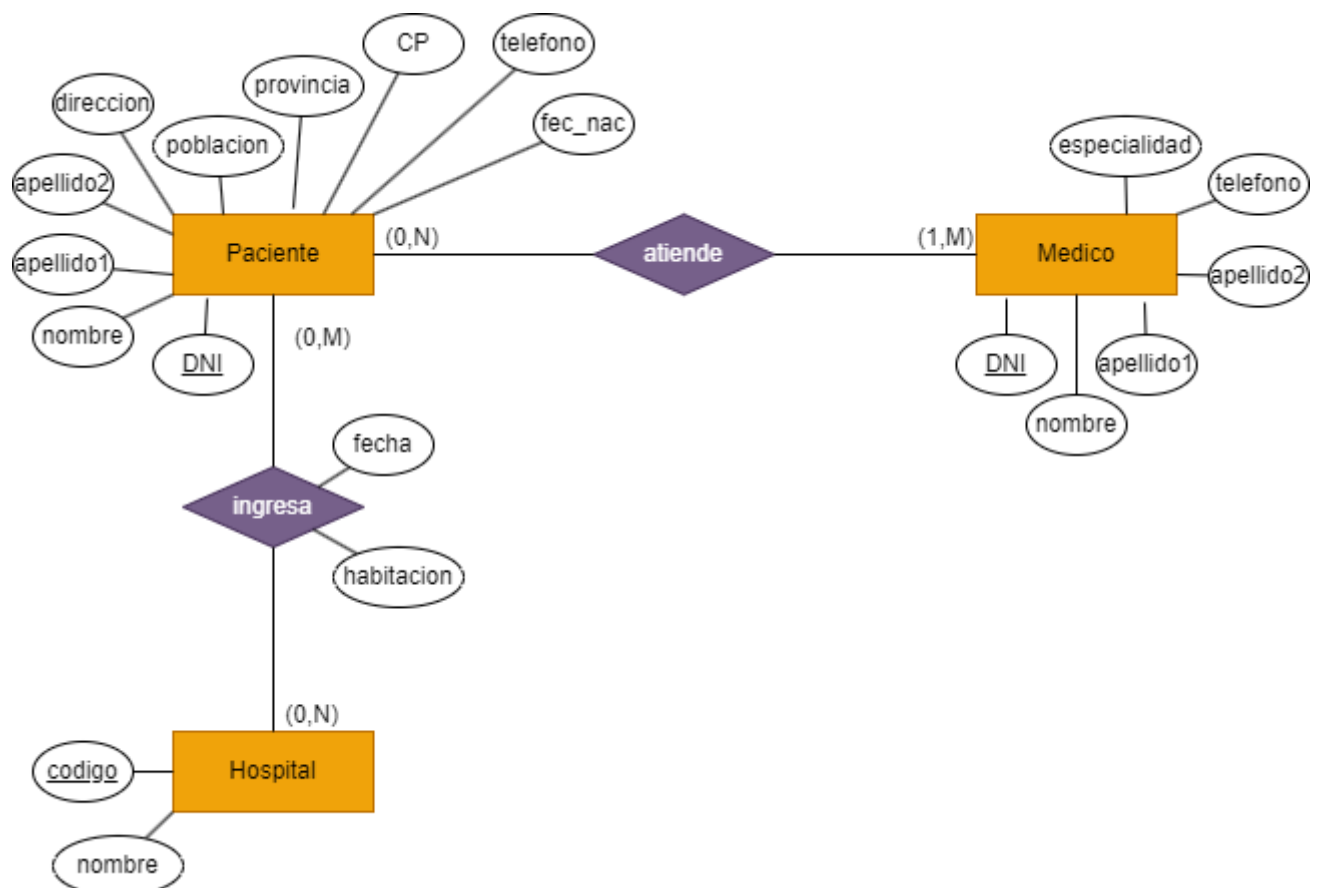
Ejercicio 06: Obtén las tablas correspondientes al siguiente Modelo Entidad/Relación mediante el método de normalización.



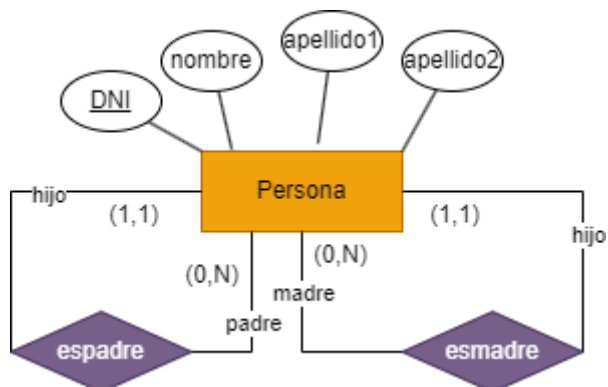
Ejercicio 07: Obtén las tablas correspondientes al siguiente Modelo Entidad/Relación mediante el método de normalización.



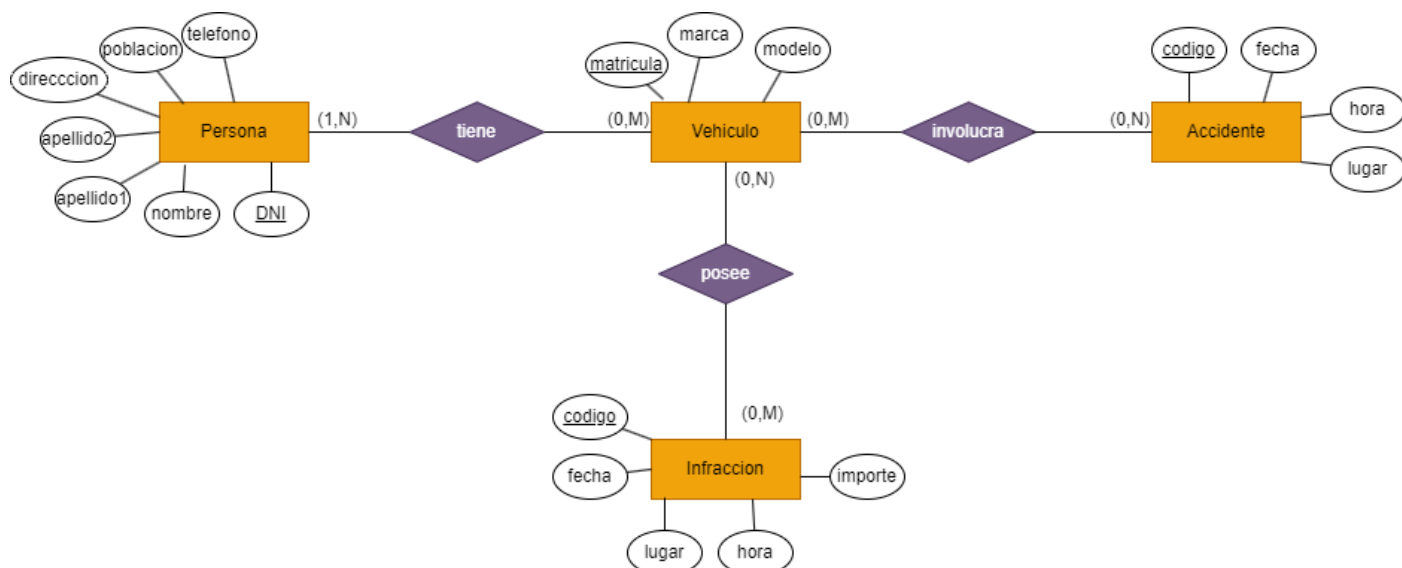
Ejercicio 08: Obtén las tablas correspondientes al siguiente Modelo Entidad/Relación mediante el método de normalización.



Ejercicio 09: Obtén las tablas correspondientes al siguiente Modelo Entidad/Relación mediante el método de normalización.



Ejercicio 10: Obtén las tablas correspondientes al siguiente Modelo Entidad/Relación mediante el método de normalización.



EJERCICIOS DE GESTIÓN DE BASES DE DATOS

Ejercicio 01: Crea una base de datos llamada TallerMecanico que será utilizada para almacenar la información de un taller mecánico.

Ejercicio 02: Crea una base de datos para la sucursal A del taller llamada TallerMecanico_SucursalA.

Ejercicio 03: El taller ha decidido cerrar la sucursal A. Elimina la base de datos TallerMecanico_SucursalA.

Ejercicio 04: Crea una base de datos para almacenar los respaldos de la información del taller llamada TallerMecanico_Respaldo.

Ejercicio 05: Renombra la base de datos TallerMecanico_Respaldo a TallerMecanico_Backup.

Ejercicio 06: Crea una base de datos para la sucursal B del taller llamada TallerMecanico_SucursalB.

Ejercicio 07: El taller ha decidido que ya no necesita la base de datos de respaldo. Elimina la base de datos TallerMecanico_Backup.

Ejercicio 08: Crea una base de datos llamada TallerMecanico_Almacen que será utilizada para gestionar el inventario de piezas.

Ejercicio 09: El taller quiere que la base de datos TallerMecanico_Almacen tenga un nombre más claro. Renombra la base de datos a TallerMecanico_Inventario.

Ejercicio 10: El taller ha decidido cerrar la sucursal B. Elimina la base de datos TallerMecanico_SucursalB.

Ejercicio 11: Comprueba mediante SQL que todas las bases de datos que has creado existen en el sistema.

EJERCICIOS DE GESTIÓN DE TABLAS

Ejercicio 01: Escribe los comandos SQL para crear una tabla de pais incluyendo las columnas idPais, nombrePais. El campo idPais será la clave principal de la tabla, de tipo numérico y deberá incrementarse automáticamente de 1 en 1. El campo nombrePais será del tipo cadena y no podrá ser nulo. La restricción de clave primaria de la tabla deberá llamarse pk_pais.

Ejercicio 02: Escribe los comandos SQL para crear una tabla region incluyendo las columnas idRegion, nombreRegion, hemisferioRegion. El campo idRegion será la clave principal de la tabla, de tipo numérico y deberá incrementarse automáticamente de 1 en 1. El campo nombreRegion será del tipo cadena y tendrá una longitud máxima de 300 caracteres. El campo hemisferioRegion solo podrá tener un valor de entre los siguientes: NORTE y SUR. La restricción de clave primaria de la tabla deberá llamarse pk_region y la restricción de los valores del campo hemisferio deberá llamarse ch_hemisferio_region.

Ejercicio 03: Modifica la tabla pais para agregarle una columna que sea la identificación de la región a la que pertenece el país, llamándose el nuevo campo idRegion. Este campo deberá referenciar al campo idPais de la tabla region y dicha restricción deberá llamarse fk_region_pais.

Ejercicio 04: Modifica la tabla pais para asegurarnos que sólo los países Italia, India y China puedan ser introducidos.

Ejercicio 05: Crea una tabla que se llame trabajo y que incluya como columnas idTrabajo, nombreTrabajo, salarioMin, salarioMax. El campo idTrabajo será del tipo cadena con una longitud máxima de 10 siendo la clave primaria, el campo nombreTrabajo será del tipo cadena con una longitud máxima de 35 no pudiendo ser nulo y los campos salarioMin y salarioMax serán del tipo numérico con decimales, teniendo 2 decimales. Se debe comprobar que el valor de salarioMax no supere el valor límite de 25000. Deberás agregar nombres a las restricciones.

Ejercicio 06: Crea una tabla que se llame trabajador y que incluya como columnas idTrabajador, nombreTrabajador, apellido1Trabajador, apellido2Trabajador y fechaNacTrabajador. El campo idTrabajador será la clave primaria y será del tipo cadena con una longitud de 9. Los campos nombreTrabajador, apellido1Trabajador y apellido2Trabajador serán del tipo cadena con una longitud de 200 y no podrán ser nulos. El campo fechaNacTrabajador deberá poder almacenar una fecha en formato 'dd/mm/yyyy' no pudiendo ser nulo. Deberás agregar nombres a las restricciones.

Ejercicio 07: Crea una tabla que se llame historial_trabajos y que incluya como columnas idTrabajador, idTrabajo, fecha_comienzo y fecha_finalización. El campo idTrabajador deberá referenciar al campo idTrabajador de la tabla trabajador y el campo idTrabajo deberá referenciar al campo idTrabajo de la tabla trabajo, ambos deberán poder borrarse y actualizarse en cascada. Los campos fecha_comienzo y fecha_finalización serán del tipo fecha y no podrán ser nulos. Crea las claves primarias que creas convenientes. Deberás agregar nombres a las restricciones.

Ejercicio 08: Modifica la tabla pais para que el nombre del país sea único.

Ejercicio 09: Modifica la tabla trabajo para que el salario mínimo tenga por defecto el valor 1500.

Ejercicio 10: Disponemos de la siguiente base de datos para gestionar la información de los pubs de una determinada provincia.

PUB
#codigo
nombre
licencia fiscal
domicilio
fecha apertura
horario
código localidad

TITULAR
#DNI titular
nombre
domicilio
código PUB

EMPLEADO
#DNI empleado
nombre
domicilio

ARTICULO
#codigo artículo
nombre
cantidad
precio
código PUB

LOCALIDAD
#código localidad
nombre

PUB_EMPLEADO
#código PUB
#DNI empleado
#función

Se pide escribir las sentencias SQL que permitan la creación de las tablas anteriores en el orden correcto teniendo en cuenta las siguientes restricciones:

- Todos los valores son de tipo carácter excepto los campos:
 - La fecha de apertura del PUB que será del tipo fecha.
 - La cantidad y el precio de las existencias y el código de localidad que serán de tipo numérico.
- Los únicos campos que no son obligatorios son los campos domicilio.
- Los valores del campo horario sólo pueden ser HORARIO1, HORARIO2 o HORARIO3.
- No es posible dar de alta EXISTENCIAS a precio 0 o negativo.
- El campo función de la tabla PUB_EMPLEADO sólo puede tener los valores CAMARERO, SEGURIDAD o LIMPIEZA.
- Se ha de mantener la integridad referencial entre las tablas.
- Las claves primarias vienen marcadas con el símbolo #.
- Debes localizar las claves foráneas.
- Todas las restricciones tienen que tener un nombre.
- Deberás nombrar los atributos de las tablas correctamente.

Ejercicio 11: La siguiente base de datos está pensada para almacenar la información necesaria para gestionar la venta automática de entradas para diferentes espectáculos desde múltiples puntos de venta, como pueden ser oficinas bancarias, terminales, o las mismas taquillas de teatros u otros recintos.

ESPECTACULO (#codESPECTACULO, nomESPECTACULO, tipoESPECTACULO, fechainicioESPECTACULO, fechafinalESPECTACULO, interpreteESPECTACULO, codRECINTO)

PRECIO_REPRESENTACION (#codESPECTACULO, #codREPRESENTACION, pvpESPECTACULO)

RECINTO (#codRECINTO, nomRECINTO, dirRECINTO, ciuRECINTO, telRECINTO, horRECINTO)

ASIENTO (#numASIENTO, #filASIENTO, codRECINTO)

REPRESENTACION (#codREPRESENTACION, codESPECTACULO, fechaREPRESENTACION, horaREPRESENTACION)

ENTRADA (#codENTRADA, codREPRESENTACION, codRECINTO, numASIENTO, filASIENTO, dniESPECTADOR)

ESPECTADOR (#dniESPECTADOR, nomESPECTADOR, dirESPECTADOR, telESPECTADOR, ciuESPECTADOR, numerotarjeta)

Se pide:

1. Justifica el orden de creación correcto de las tablas.
2. Establecer las claves primarias de cada una de las tablas y las restricciones de integridad referencial existentes entre las mismas (claves foráneas).
3. Decidir y justificar el tipo de dato de cada uno de los atributos de las tablas.
4. Crear las sentencias SQL que nos permitan crear las tablas anteriores, debiendo haber como mínimo una de cada de las siguientes restricciones:
 - a) NOT NULL.
 - b) UNIQUE.
 - c) CHECK.
 - d) DEFAULT.
 - e) IDENTITY.
5. Todas las restricciones tienen que tener un nombre.
6. Deberás nombrar los atributos de las tablas correctamente.

Ejercicio 12: De cada uno de los ejercicios de la relación de una unidad didáctica 02 se pide:

1. Paso a tablas del modelo entidad relación obtenido, normalizando hasta FNBC.
2. Escribir las sentencias SQL que permitan la creación de las tablas. Justifica el orden de creación correcto de las tablas.
3. Establecer las claves primarias de cada una de las tablas y las restricciones de integridad referencial existentes entre las mismas.
4. Decidir y justificar el tipo de dato de cada uno de los atributos de las tablas.
5. Crear restricciones de los siguientes tipos:
 - a) NOT NULL.
 - b) UNIQUE.
 - c) CHECK.
 - d) DEFAULT.
 - e) IDENTITY.
6. Todas las restricciones tienen que tener un nombre.
7. Deberás nombrar los atributos de las tablas correctamente.

EJERCICIOS DE GESTIÓN DE USUARIOS

La base de datos de una biblioteca tiene las siguientes tablas principales:

- libro: Información de los libros disponibles.
- prestamo: Registro de los préstamos de libros.
- usuario: Información de los usuarios de la biblioteca.
- autor: Información de los autores de los libros.

Para la base de datos de la biblioteca los siguientes ejercicios.

Ejercicio 01: Crear un usuario "bibliotecario1" con un login y una contraseña.

Ejercicio 02: Crear un usuario "lector1" sin login.

Ejercicio 03: Modificar el nombre del usuario "bibliotecario1" a "gestorBiblioteca".

Ejercicio 04: Asignar permisos de SELECT y UPDATE sobre la tabla libro al usuario "gestorBiblioteca".

Ejercicio 05: Denegar permisos de DELETE sobre la tabla libro al usuario "gestorBiblioteca".

Ejercicio 06: Asignar permisos de SELECT sobre la tabla prestamo al usuario "lector1".

Ejercicio 07: Crear un nuevo usuario "bibliotecario2" con login y contraseña, pero denegarle permisos de UPDATE sobre la tabla libro.

Ejercicio 08: Asignar al usuario "gestorBiblioteca" permisos para insertar, modificar y borrar datos de la tabla prestamo y hacer que pueda conceder esos permisos a otros usuarios.

Ejercicio 09: Asignar todos los permisos sobre la tabla usuario al usuario "gestorBiblioteca".

Ejercicio 10: Eliminar el permiso de SELECT sobre la tabla prestamo del usuario "lector1".

Ejercicio 11: Renombrar al usuario "bibliotecario2" a "asistenteBiblioteca" y cambiar su login.

Ejercicio 12: Crear un nuevo usuario "revisorLibros" que pueda realizar operaciones de consulta y actualización solo sobre la tabla libro.

Ejercicio 13: Eliminar al usuario "lector1" y el login asociado.

Ejercicio 14: Eliminar todos los permisos del usuario "asistenteBiblioteca" sobre la tabla libro.