**UD3 – EJERCICIO 1 – EJERCICIOS MODELO E-R**

[EJERCICIOS BÁSICOS DE E-R: DIAGRAMAS 2](#_Toc127091355)

[EJERCICIOS BÁSICOS DE E-R: PASO A TABLAS 17](#_Toc127091356)

[EJERCICIOS DE E-R EXTENDIDO: DIAGRAMAS 26](#_Toc127091357)

[EJERCICIOS DE E-R EXTENDIDO: PASO A TABLAS 32](#_Toc127091358)

[EJERCICIOS DE E-R: NORMALIZACIÓN DE TABLAS 35](#_Toc127091359)

# EJERCICIOS BÁSICOS DE E-R: DIAGRAMAS

1. Diagrama

   Descripción generada automáticamenteEn una academia los profesores dan clase a los alumnos matriculados de modo que todo profesor da clase al menos a un alumno y todo alumno recibe clase de un único profesor. No es necesario introducir atributos.

* Un profesor enseña a mínimo 1 alumno y a máximo n alumnos.
* Un alumno es enseñado por mínimo 1 profesor y máximo 1 profesor.
* Relación: 1:N (1 a muchos).

1. Diagrama

   Descripción generada automáticamenteLos profesores de un centro pueden ser o no tutores de un alumno, en cualquier caso, los alumnos solo podrán tener un único tutor. No es necesario introducir atributos.

* Un profesor es tutor de mínimo 0 alumnos y máximo n alumnos.
* Un alumno es tutorizado por mínimo 1 profesor y máximo 1 profesor.
* Relación: 1:N (1 a muchos).

1. Representa la relación entre Personas y sus padres (masculino). No es necesario introducir atributos.

Forma

Descripción generada automáticamente con confianza media

* Una persona es padre de mínimo 0 personas y máximo n personas.
* Una persona tiene como padre a mínimo 1 persona y máximo 1 persona.
* Relación: 1:N (1 a muchos).

1. En un centro de salud un médico atiende a varios pacientes y cada paciente está asignado a un solo médico. Cada médico pasa consulta en una o varias salas en horario distinto. En cada sala pasan consulta varios médicos.
   * Del médico interesan conocer sus datos personales (nombre y dirección) y el año en que se colegió.
   * De la sala interesa conocer un nº de sala.
   * Del paciente, se desea conocer su historial médico, al que se accede a través del número de la Seguridad Social.
   * Diagrama

     Descripción generada automáticamenteSe desea saber en qué sala pasa consulta cada médico en cada momento.

RELACIÓN MÉDICO – TRATA – PACIENTE

* Un médico trata a mínimo 1 paciente y máximo n pacientes.
* Un paciente es tratado por mínimo 1 médico y máximo 1 médico.
* Relación: 1:N (1 a muchos).

RELACIÓN MÉDICO – UTILIZA – SALA

* Un médico utiliza mínimo 1 sala y máximo n salas.
* Una sala es utilizada por mínimo 1 médico y máximo n médicos.
* Relación: N:M (muchos a muchos).

1. Se desea construir una bd para mantener información sobre los equipos y partidos de la liga. Un equipo tiene un cierto número de jugadores y no todos participan en cada partido. Queremos registrar, por cada partido, qué jugadores juegan y en qué posición, la fecha del partido y el resultado.

Para identificar un equipo se emplea su nombre, y además se conoce el año en que se federó. Para identificar a un jugador se emplea un identificador de jugador (número), y además se conoce su nombre y teléfono.

Diagrama

Descripción generada automáticamentePara identificar a un partido se emplea un identificador de partido.

RELACIÓN EQUIPO – TIENE – JUGADOR:

* Un equipo tiene mínimo n jugadores y máximo m jugadores.
* Un jugador pertenece a mínimo 1 equipo y máximo 1 equipo.
* Relación: 1:N (1 a muchos).

RELACIÓN JUGADOR – JUEGA – PARTIDO:

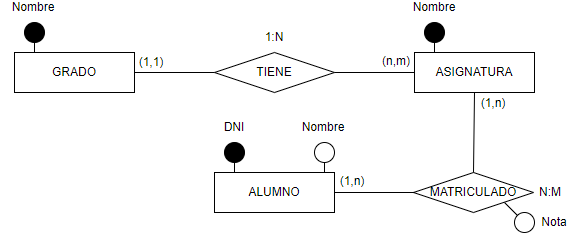
* Un jugador juega un mínimo de 0 partidos y máximo n partidos.
* Un partido es jugado por mínimo n jugadores y máximo m jugadores.
* Relación: N:M (muchos a muchos).

1. Se desea diseñar una BD para una Universidad que contenga información sobre grados (carreras) y asignaturas que se pueden estudiar.

Además, se incluirá la información de los alumnos matriculados en las correspondientes asignaturas y las calificaciones que obtienen en las asignaturas en las que están matriculados.

Introducir atributos básicos para cada entidad.

Nota: suponer que una asignatura solo puede pertenecer a un grado.



RELACIÓN GRADO – TIENE – ASIGNATURA

* Un grado tiene un mínimo de n asignaturas y un máximo de m asignaturas.
* Una asignatura pertenece a mínimo 1 grado y máximo 1 grado.
* Relación: 1:N (1 a muchos).

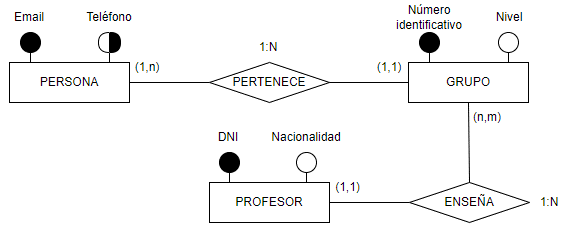
RELACIÓN ASIGNTURA – MATRICULADO – ALUMNO

* Una asignatura tiene un mínimo de 1 alumno matriculado y un máximo de n alumnos matriculados.
* Un alumno está matriculado en mínimo 1 asignatura y un máximo de n asignaturas.
* Relación: N:M (muchos a muchos).

1. Una persona puede asistir a clases particulares de inglés en una academia.

Cada persona se ubica en un grupo en función de su nivel de inglés. Cada grupo se identifica por un número y tiene un nivel determinado.

En la academia hay varios profesores de nacionalidades distintas, cada profesor da clase a varios grupos. A cada grupo solo le imparte clase un profesor, identificado por su DNI.

Para localizar a un alumno se le puede llamar por teléfono o se le puede enviar un correo electrónico. Aparte de eso solo se conoce su nombre completo (que puede repetirse en distintos alumnos).

RELACIÓN PERSONA – PERTENECE – GRUPO

* Una persona pertenece a mínimo 1 grupo y a máximo 1 grupo.
* Un grupo está formado por mínimo 1 persona y máximo n personas.
* Relación: 1:N (1 a muchos).

RELACIÓN PEROFESOR – ENSEÑA – GRUPO

* Un profesor enseña a mínimo n grupos y máximo m grupos.
* Un grupo es enseñado por mínimo 1 profesor y máximo por 1 profesor.
* Relación: 1:N (1 a muchos).

1. En un taller mecánico se quiere guardar la información de los empleados que trabajan. Para cada uno se conoce su DNI, nombre, teléfono y función dentro del taller. Un empleado puede realizar diversas funciones.

En el taller hay distintas funciones que realizar, cada una con un nombre distinto y una remuneración.

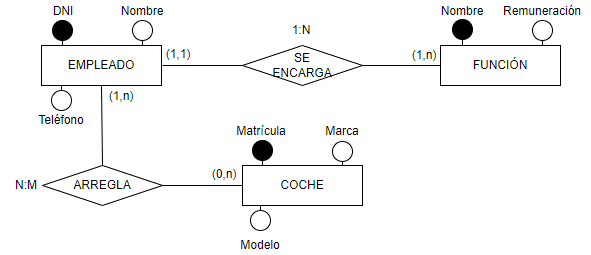
Cada una es realizada por un único trabajador.

También se quiere guardar información de los coches que repara cada empleado. A veces en la reparación de un coche se requiere que participen varios empleados.

Algunos empleados administrativos no reparan coches.

De cada coche se conoce su matrícula, modelo y año de fabricación.

Establece el modelo entidad/relación con las especificaciones dadas.



RELACIÓN EMPLEADO – SE ENCARGA – FUNCIÓN

* Un empleado se encarga de mínimo 1 función y máximo n funciones.
* Una función puede ser realizada por mínimo 1 trabajador y máximo 1 trabajador.
* Relación: 1:N (uno a muchos).

RELACIÓN EMPLEADO – ARREGLA – COCHE

* Un empleado arregla un mínimo de 0 coches y un máximo de n coches.
* Un coche es arreglado por un mínimo de 1 empleado y máximo de n empleados.
* Relación: N:M (muchos a muchos).

1. Un profesor de historia de la Europa medieval quiere tener una base de datos con la información referente a cruzadas de la Edad Media. Para el diseño de dicha base de datos se cuenta con la siguiente especificación de necesidades de información:

* Caballeros: interesa saber su nombre y su fecha de nacimiento. También interesa conocer de quien son hijos, sabiendo que, según las normas sociales de la época, los caballeros solo pueden ser hijos de caballeros. Es habitual que un caballero hijo y su padre pueden tener el mismo nombre. No se conocen los apellidos de los caballeros. También interesa conocer en qué provincia nacieron, en qué provincia gobernaron y en que guerras (cruzadas) participaron bajo las órdenes de qué rey.
* Provincia: interesa almacenar el número de habitantes, el nombre de la provincia, los caballeros que la han gobernado y los que han nacido allí. Una provincia puede ser gobernada por varios caballeros (como es lógico, en fechas distintas) y un caballero puede gobernar en varias provincias. Interesa conocer la fecha de comienzo y final de gobierno de cada caballero sobre cada provincia.
* Cruzada: interesa saber su nombre, contra quien fue la cruzada y su resultado, así como los reyes y caballeros participantes. En una cruzada puede participar más de un rey. Cualquier caballero y cualquier rey pueden participar en cualquier cruzada.
* Reyes: participan en cruzadas a través de caballeros. Se conoce el país y fecha de nacimiento.

Diagrama

Descripción generada automáticamenteRealizar el modelo de entidad-relación correspondiente utilizando la notación de Chen.

RELACIÓN CABALLERO – ES HIJO – CABALLERO

* Un caballero es hijo de mínimo 1 caballero y máximo 1 caballero.
* Un caballero es padre de mínimo 0 caballeros y máximo n caballeros.
* Relación: 1:N (uno a muchos).

RELACIÓN CABALLERO – NACE – PROVINCIA

* Un caballero nace en mínimo 1 provincia y máximo 1 provincia.
* En una provincia nacen un mínimo de 1 caballeros y máximo n caballeros.
* Relación: 1:N (uno a muchos).

RELACIÓN CABALLERO – GOBIERNA – PROVINCIA

* Un caballero gobierna en mínimo 1 provincia y máximo n provincias.
* En una provincia gobiernan un mínimo de 1 caballero y un máximo de n caballeros.
* Relación: N:M (muchos a muchos).

RELACIÓN CABALLERO / REY – PARTICIPA – CRUZADA (relación ternaria)

* Cualquier caballero y cualquier rey participan en mínimo 1 cruzada y máximo n cruzada.
* Cualquier rey y cualquier cruzada es librada por mínimo 1 caballero y máximo n caballeros.
* Cualquier cruzada y cualquier caballero tienen la partición de un mínimo de 1 rey y un máximo de n.
* Relación: N:M:N (muchos a muchos a muchos).

1. Se desea diseñar una Base de Datos que almacene información sobre la red de metro. Se tendrán en cuenta los siguientes supuestos:

* Los trenes están asignados a una línea determinada.
* Cada uno se guarda en una cochera determinada.
* En una línea, cada estación tiene un número de orden.

Vamos que tener almacenados los siguientes datos: El número y nombre de cada línea. El código de tren y su tipo. El código de la cochera y su capacidad. El código de la estación y el orden en el que se encuentra para cada línea que por ella pase.

Diagrama

Descripción generada automáticamente

RELACIÓN TREN – ASIGNADO – LÍNEA

* Un tren está asignado a mínimo 1 línea y máximo 1 línea.
* A una línea están asignados un mínimo de 1 tren y máximo de n trenes.
* Relación: 1:N (uno a muchos).

RELACIÓN TREN – SE GUARDA – COCHERA

* Un tren se guarda en mínimo 1 cochera y máximo 1 cochera.
* En una cochera se guardan un mínimo de 1 tren y un máximo de n trenes.
* Relación: 1:N (uno a muchos).

RELACIÓN LÍNEA – ESTÁ COMPUESTA – ESTACIÓN

* Una línea está compuesta de mínimo n estaciones y máximo m estaciones.
* Una estación es parte de mínimo 1 línea y máximo n líneas.
* Relación: N:M (muchos a muchos).

1. Se desea diseñar una Base de Datos para un banco según los siguientes supuestos:

* El banco tiene distintas sucursales que se identifican por un código y están en una dirección.
* Cada sucursal tiene una serie de cuentas corrientes asignadas a ella que se identifican por un código, distinto para cada cuenta, aunque sea de distinta sucursal. Una cuenta corriente tiene uno o varios clientes como titulares.
* Por otra parte, cada cliente, que se identifica por su carné de identidad, puede tener varias cuentas, y por supuesto unos privilegios distintos, en cada una de ellas. Se considera que una persona es cliente del banco si tiene al menos una cuenta abierta en él.
* Los clientes pueden realizar distintas operaciones sobre las cuentas del banco (por ejemplo “ingreso”, “transferencia”, “solicitud tarjeta”). Es posible que un cliente pueda realizar operaciones sobre cuentas de las que no es titular (por ejemplo, un ingreso). Una misma operación puede afectar a una o varias cuentas, tiene una descripción y se identifica por un código. Cada operación es realizada por un único cliente y se guarda la fecha y hora de realización.

Ejemplo: un cliente con DNI 45676054R puede realizar la operación de código 23 y descripción “transferencia” el día 23-05-2020 a las 13:54 sobre las cuentas 6000245 y 6000987. Sólo es titular de la primera.

* Cada cuenta puede tener domiciliaciones asociadas con ellas, las cuales se identifican con un código.
* Diagrama

  Descripción generada automáticamenteCada cliente puede tener otorgados prestamos, sin que estos préstamos estén asociados a ninguna de las cuentas corrientes. Cada préstamo se identifica por un número, se otorga a nombre de un solo cliente y tiene una cantidad, fecha de contratación e interés TAE.

RELACIÓN SUCURSAL – TIENE – CUENTA CORRIENTE

* Una sucursal tiene un mínimo de 1 cuenta corriente y un máximo de n cuentas.
* Una cuenta pertenece a un mínimo de 1 sucursal y un máximo de 1 sucursal.
* Relación: 1:N (uno a muchos).

RELACIÓN CUENTA CORRIENTE – ASOCIA – DOMICILIACIÓN

* Una cuenta corriente tiene asociados un mínimo de 0 domiciliaciones y un máximo de n domiciliaciones.
* Una domiciliación pertenece a un mínimo de 1 cuenta y un máximo de 1 cuenta.
* Relación: 1:N (uno a muchos).

RELACIÓN CUENTA CORRIENTE – PERTENECE – CLIENTE

* Una cuenta pertenece a un mínimo de 1 cliente y un máximo de n clientes.
* Un cliente tiene un mínimo de 1 cuenta y un máximo de n cuentas.
* Relación: 1:N (uno a muchos).

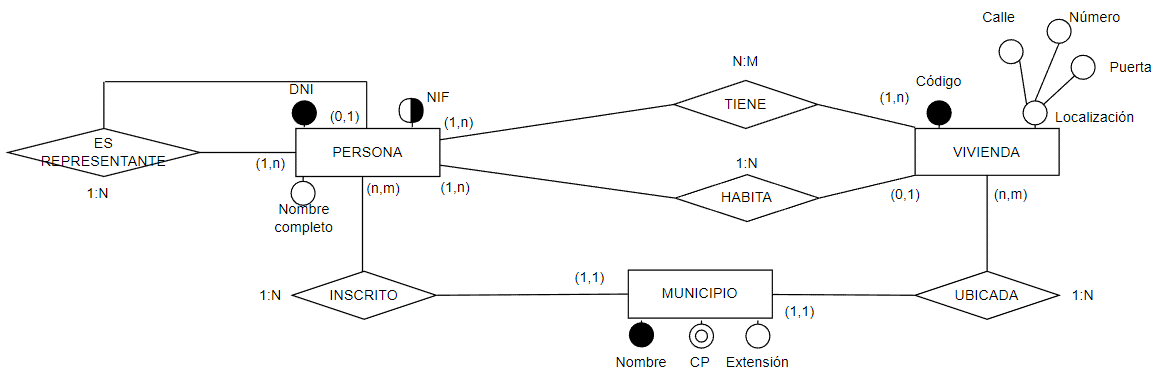
RELACIÓN CLIENTE – TIENE OTORGADO – PRÉSTAMO

* Un cliente tiene un mínimo de 0 préstamos y un máximo de n préstamos.
* Un préstamo es de mínimo 1 cliente y máximo 1 cliente.
* Relación: 1:N (uno a muchos).

RELACIÓN CLIENTE / CUENTA CORRIENTE / OPERACIÓN – REALIZA (relación ternaria)

* Un cliente con una cuenta puede realizar un mínimo de 1 operación y máximo de n operaciones.
* Un cliente realiza una operación en mínimo 1 cuenta y máximo en n cuentas.
* Una operación en una cuenta es realizada por mínimo de 1 cliente y máximo de n clientes.
* Relación: N:M:N (muchos a muchos a muchos).

1. Supongamos un sistema informático en el que queremos tener información de municipios, viviendas y personas. Cada persona solo puede habitar en una vivienda y en cambio puede ser propietaria de más de una vivienda.

* En cada unidad familiar (personas que viven en el mismo domicilio) se identifica una persona como representante.
* De cada persona se va a guardar el DNI, el nombre, los apellidos, NIF.
* De los municipios nos interesa saber su código postal (uno o varios), su nombre (identificativo) y su extensión.
* De cada vivienda vamos a guardar el código de la vivienda, la dirección completa (estará formada por la calle, número y la puerta). También interesa conocer el municipio donde está ubicada.
* Una persona puede estar inscrita en un municipio distinto al que está su vivienda habitual.

RELACIÓN PERSONA – ES REPRESENTANTE – PERSONA

* Una persona es representante de mínimo 1 persona y máximo n.
* Una persona es representada por mínimo 0 persona y máximo 1.
* Relación: 1:N (uno a muchos).

RELACIÓN PERSONA – TIENE – VIVIENDA

* Una persona tiene un mínimo de 1 vivienda y máximo de n viviendas.
* Una vivienda es posesión de mínimo 1 persona y máximo n personas.
* Relación: N:M (muchos a muchos).

RELACIÓN PERSONA – HABITA – VIVIENDA

* Una persona habita en mínimo 0 viviendas y máximo 1 vivienda.
* Una vivienda es habitada por un mínimo de 1 persona y un máximo de n personas.
* Relación: 1:N (uno a muchos).

RELACIÓN VIVIENDA – UBICADA – MUNICIPIO

* Una vivienda está ubicada en 1 municipio.
* Un municipio ubica un mínimo de n viviendas y un máximo de m viviendas.
* Relación: 1:N (uno a muchos).

RELACIÓN PERSONA REPRESENTANTE – INSCRITO – MUNICIPIO

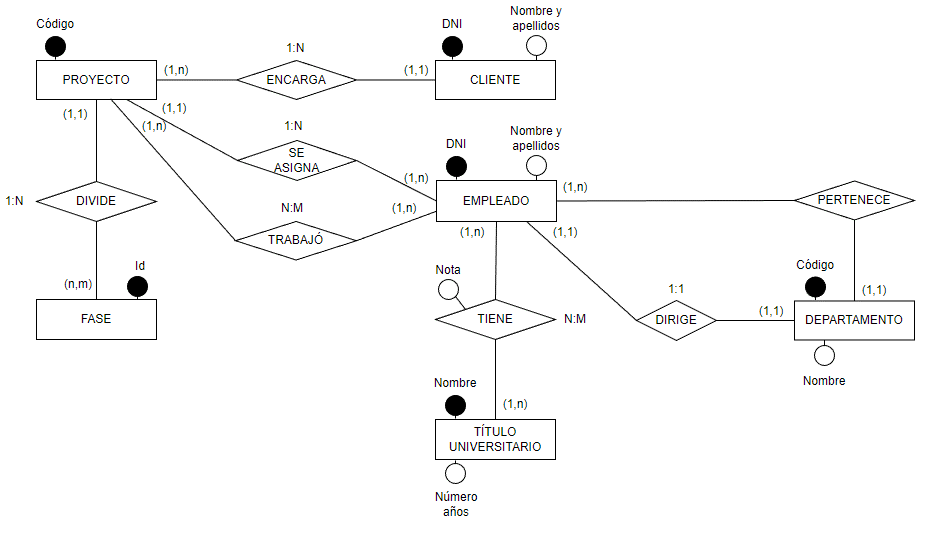
* Una persona está inscrita en 1 municipio.
* Un municipio tiene inscritas a un mínimo de n personas y máximo de m personas.
* Relación: 1:N (uno a muchos).

1. Se quiere diseñar un modelo de datos de una empresa que se dedica a la elaboración de proyectos. Estos proyectos son encargados a la empresa por clientes externos. Cada proyecto es encargado por un único cliente. Cada uno de los proyectos está dividido en fases, y de cada una de las fases interesa conservar información relativa a su desarrollo.

Dentro de la empresa existen una serie de empleados, y cada uno de ellos está asignado a un único proyecto en la actualidad. Sin embargo, a lo largo de su vida profesional, un empleado ha trabajado en muchos proyectos (se quiere guardar esa información).

Los empleados son todos titulados universitarios, con diversas carreras. Se desea conocer el número de años que dura cada carrera y el nivel que tiene.

La empresa está dividida en departamentos, donde cada uno tiene un único jefe al que se le asignan varios empleados. Un empleado no puede pertenecer simultáneamente a más de un departamento.

Establece atributos básicos.

RELACIÓN CLIENTE – ENCARGA – PROYECTO

* Un cliente encarga un mínimo de 1 proyecto y máximo de n.
* Un proyecto es encargado por 1 cliente.
* Relación: 1:N (uno a muchos).

RELACIÓN PROYECTO – DIVIDE – FASE

* Un proyecto se divide en un mínimo de n fases y un máximo de m fases.
* Una fase divide a 1 proyecto.
* Relación: 1:N (uno a muchos).

RELACIÓN PROYECTO – SE ASIGNA – EMPLEADO

* Un empleado se asigna a 1 proyecto.
* Un proyecto es asignado mínimo 1 empleado y máximo n empleados.
* Relación: 1:N (uno a muchos).

RELACIÓN PROYECTO – TRABAJÓ – EMPLEADO

* Un empleado ha trabajado en mínimo 1 proyecto y máximo n proyectos.

Un proyecto ha sido trabajado por mínimo 1 empleado y máximo n empleados.

* Relación: N:M (muchos a muchos).

RELACIÓN EMPLEADO – TIENE – TÍTULO UNIVERSITARIO

* Un empleado tiene un mínimo de 1 título y un máximo de n.
* Un título universitario es posesión de un mínimo de 1 empleado y un máximo de m.
* Relación: N:M (muchos a muchos).

RELACIÓN EMPLEADO – TRABAJA – DEPARTAMENTO

* Un empleado trabaja en 1 departamento.
* En un departamento trabaja mínimo 1 empleado y máximo n empleados.
* Relación: 1:N (uno a muchos).

RELACIÓN EMPLEADO – DIRIGE – DEPARTAMENTO

* Un empleado dirige 1 departamento.
* Un departamento es dirigido por 1 empleado.
* Relación: 1:1 (uno a uno).

1. Un taller de reparaciones de vehículos desea informatizar su negocio para lo cual proporciona al analista encargado del desarrollo la siguiente información de necesidades y hechos:

* Las averías incluyen: descripción de la avería, pieza a cambiar (única en cada avería) e importe de la reparación. Todas las averías tienen una pieza a sustituir.
* Los clientes se tienen un número de cliente, nombre y apellidos, matrícula del vehículo a reparar (única) y número de teléfono del cliente.
* Al cliente se le factura, en una factura, la reparación completa, que puede incluir varios partes de avería (puede haber partes de avería que no se incluyan en la factura, ya que el cliente puede no haberlas aceptado). Un mismo cliente puede tener varias facturas.
* Las nuevas piezas que se sustituyen en las averiadas están almacenadas en el taller, gracias al suministro de los proveedores correspondientes. Estos son identificados por su código de proveedor, y además se conoce su teléfono y categoría. Las piezas tienen un número identificativo, descripción, precio y stock disponible.
* Dos piezas pueden tener el mismo número identificativo, siempre que tengan distintas descripciones. Por ejemplo, pueden existir la pieza de número 45 y descripción “tornillo” y, por otro lado, la pieza de número 45 y descripción “tuerca”. Se consideran piezas distintas.
* Una pieza puede tener un precio en el almacén (aplicable a las averías) distinto al precio de suministro del proveedor.
* Interesa conocer por cada Proveedor y pieza suministrada el número de unidades y el precio por unidad.

Diagrama

Descripción generada automáticamente

RELACIÓN FACTURA – INCLUYE – PARTE AVERÍA

* Una factura incluye un mínimo de 1 parte de avería y un máximo de n.
* Un parte de avería está incluido un mínimo de 0 facturas y máximo de 1.
* Relación: 1:N (uno a muchos).

RELACIÓN FACTURA – SE ENTREGA – CLIENTE

* Una factura es entregada a 1 cliente.
* Un cliente recibe un mínimo de 1 factura y un máximo de n.
* Relación: 1:N (uno a muchos).

RELACIÓN PARTE AVERÍA – TIENE – PIEZA SUSTITUCIÓN

* Un parte de avería tiene 1 pieza de sustitución.
* Una pieza de sustitución pertenece a un mínimo de 0 parte de avería y un máximo de n partes de avería.
* Relación: 1:N (uno a muchos).

RELACIÓN PIEZA SUSTITUCIÓN – SUMINISTRADA – PROVEEDOR

* Una pieza de sustitución es suministrada por un mínimo de 1 proveedor y un máximo de n.
* Un proveedor suministra un mínimo de 1 pieza de sustitución y un máximo de n.
* Relación: N:M (muchos a muchos).

# EJERCICIOS BÁSICOS DE E-R: PASO A TABLAS

1. Haz el paso a tablas de los siguientes casos:



REVISTA (**COD**, sala, estantería)



ACTOR (**DNI**, nombre\_artístico)

PREMIOS (**dni\_actor, premio)**

FK: dni\_actor referencia ACTOR(DNI) borrado CASCADE



PERSONA (**DNI**, nombre)

ES HIJO (**dni\_padre, dni\_hijo)**

FK: dni\_padre referencia PERSONA (DNI) borrado CASCADE

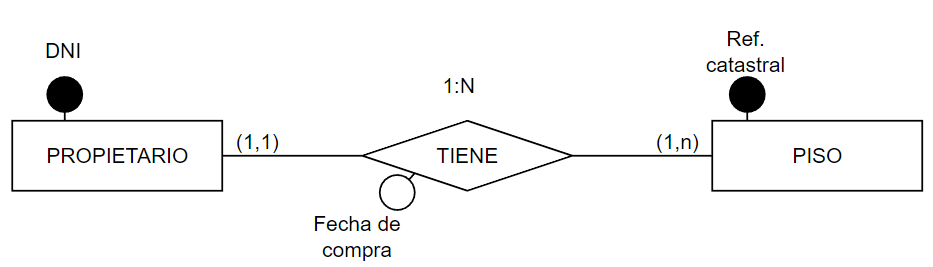
FK: dni\_hijo referencia PERSONA (DNI) borrado CASCADE

1. Crea el modelo Entidad/Relación y después haz el paso a tablas.

Un piso siempre es de un propietario. Un propietario puede tener uno o más pisos.

Incluye los atributos: referencia catastral del piso, DNI del propietario, fecha de la compra.

Si se da de baja a un propietario, sus pisos quedarán en la BD sin propietario asignado.



PROPIETARIO (**DNI)**

PISO (**Ref. catastral**, fecha de compra, **dni\_propietario)**

FK: dni\_propietario referencia PROPIETARIO (DNI) borrado SET NULL.

1. Crea el modelo Entidad/Relación y después haz el paso a tablas. Consideramos dos entidades: persona y huella dactilar, con una relación entre ellas. La clave de persona es DNI y de huella es codificación.
2. A cada persona se le toma la huella dactilar de un sólo dedo, ya que se considera suficiente para identificarla. Toda persona tiene una huella tomada.

Diagrama

Descripción generada automáticamente

PERSONA (**DNI**)

HUELLA (**COD**, **dni\_persona**)

FK: dni\_persona referencia PERSONA (DNI) borrado CASCADE.

1. Al supuesto anterior se añade el hecho de que hay personas que no tienen manos o dedos.

Diagrama

Descripción generada automáticamente

PERSONA(**DNI**)

HUELLA (**COD**, **dni\_persona**)

FK: dni\_persona referencia PERSONA (DNI) borrado CASCADE.

1. A lo anterior se añade que habrá huellas de prueba generadas por ordenador, que no pertenecerán a ninguna persona.

Diagrama

Descripción generada automáticamente

PERSONA (**DNI**)

HUELLA (**COD**)

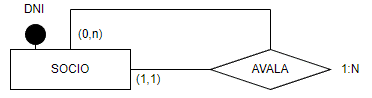
TIENE (**dni\_persona, cod\_huella**)

FK: dni\_persona referencia PERSONA (DNI) borrado CASCADE.

FK: cod\_huella referencia HUELLA (COD) borrado CASCADE.

1. Crea el modelo Entidad/Relación y después haz el paso a tablas. Se trata de un club de socios.
2. Cada socio tiene obligatoriamente otro socio que le ha avalado (avalista) para poder hacerse miembro del club. Un socio puede avalar o no a otros socios.

Un socio que ha avalado a otros no puede ser dado de baja del club, mientras sus avalados estén en el club.



SOCIO (**DNI**, **dni\_avalista**)

FK: dni\_avalista referencia SOCIO (DNI) borrado RESTRICT

1. Un socio no tiene por qué ser avalado, de hecho, pocos serán avalados. Un socio puede avalar o no a otros socios.

Un socio que ha avalado a otros no puede ser dado de baja del club, mientras sus avalados estén en el club.

Imagen que contiene Forma

Descripción generada automáticamente

SOCIO (**DNI**)

AVALA (**dni\_avalista,** **dni\_avalado**)

FK: dni\_avalista referencia SOCIO (DNI) borrado CASCADE

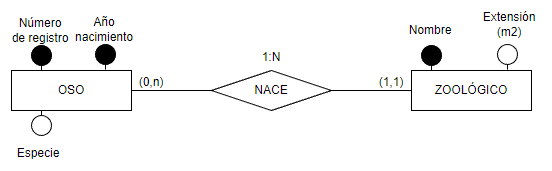
FK: dni\_avalado referencia SOCIO (DNI) borrado RESTRICT

1. Crea el modelo Entidad/Relación y después haz el paso a tablas. Se trata de una BD de osos pardos nacidos en zoológicos.

Un oso se identifica por un número de registro y su año de nacimiento. Nace en un zoológico, el cual se identifica por un nombre. En un zoo pueden nacer o no osos. También se desea conocer la especie de los osos (obligatoriamente).

De cada zoológico se conoce su extensión en metros cuadrados.

No puede darse de baja en la BD un zoológico si han nacido osos en ella.



ZOOLÓGICO (**nombre**, extensión)

OSO (**número\_registro, año\_nacimiento**, especie, **nombre\_zoológico**)

NOT NULL: especie

FK: nombre\_zoológico referencia ZOOLÓGICO (nombre) borrado RESTRICT

1. Crea el modelo Entidad/Relación y después haz el paso a tablas de un caso de compra de artículos por parte de clientes.

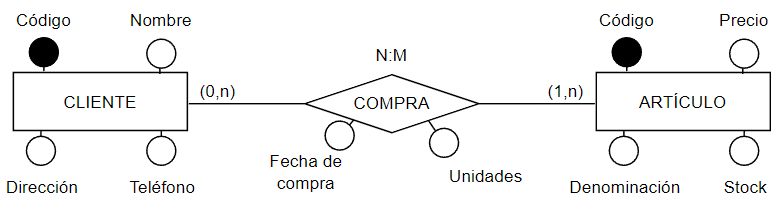
Un cliente se identifica por un código, además tiene nombre, dirección y teléfono.

El cliente puede comprar uno o varios artículos, identificados por un código. Los artículos tienen además un precio, denominación y stock.

Un artículo puede ser comprado por un cliente, varios o ninguno. En caso de que haya compra, interesa saber la fecha y las unidades que se compraron del artículo.

Si se da de baja a un cliente, automáticamente se darán de baja en la BD todas sus compras realizadas.

Si se da de baja un artículo que ha sido comprado en alguna ocasión por algún cliente, en la base de datos quedará reflejado que ese cliente compró un artículo “descatalogado”.



CLIENTE(**código**, nombre, dirección, teléfono)

ARTÍCULO(**código**, denominación, precio, stock)

COMPRA(**código\_cliente, código\_artículo**, fecha\_compra, unidades)

FK: código\_cliente referencia CLIENTE (código) borrado CASCADE

FK: código\_artículo referencia ARTÍCULO (código) borrado SET DEFAULT “descatalogado”

1. Crea el modelo Entidad/Relación y después haz el paso a tablas:

Un libro está clasificado en un único tema. Cada tema puede tener asociados varios libros (o ninguno). Del tema interesa conocer el código y descripción.

De cada libro interesa conocer su código identificativo, autor, ISBN, título y número de ejemplares (autor y título son obligatorios).

No puede darse de baja un tema si hay libros en ese tema.

Diagrama

Descripción generada automáticamente

TEMA (**código**, descripción)

LIBRO (**Código\_identificativo**, ISBN, autor, título, número\_ejemplares, **código\_tema**)

NOT NULL: ISBN, autor, título

UNIQUE: ISBN

FK: código\_tema referencia TEMA (código) borrado RESTRICT

1. Crea el modelo Entidad/Relación y después haz el paso a tablas del caso de una empresa:

Un empleado de una empresa puede dirigir a otro empleado y tiene un único director. Algunos empleados (director general, comerciales, etc) no tienen a nadie que les dirija.

Para cada empleado se conoce su código de empleado, dirección, teléfono y nombre.

Si se despide a un empleado que es director de otros empleados, esos empleados se quedarán sin director asignado.

Diagrama

Descripción generada automáticamente

EMPLEADO (**Código**, nombre, dirección, teléfono)

ES\_DIRECTOR (**código\_director**, **código\_subordinado**)

FK: código\_director referencia EMPLEADO(código) borrado SET NULL

FK: código\_subordinado referencia EMPLEADO (código) borrado CASCADE

1. Haz el paso a tablas del siguiente modelo de un taller de reparación de vehículos. Ten en cuenta que:

* Una avería puede o no ser aceptada por un cliente. En caso de que sea aceptada, formará parte, junto con otras averías, de una factura a nombre de ese cliente. Puede haber muchas que no se acepten por clientes.
* Si se borra un cliente de la BD, para no perder información fiscal, no se deben borrar las facturas a su nombre, sino que se quedarán guardadas sin cliente asignado.
* No es posible borrar una pieza de la BD si se ha referenciado en alguna avería.
* Si se borra un proveedor de la BD, las piezas que suministraba ese proveedor se quedarán automáticamente con el código de proveedor ficticio “00”.



PROVEEDOR (**cod\_prov**, tlfno, categ)

PIEZA (**num\_pieza**, descr, stock, precio)

SUMINISTRA (**proveedor, pieza**, precio, unidades)

FK: proveedor referencia PROVEEDOR (cod\_prov) borrado SET DEFAULT “00”

FK: pieza referencia PIEZA (num\_pieza) borrado CASCADE

CLIENTE (**num\_cli**, nombre, apell, matric, tlfno)

FACTURA (**num\_fac**, total, **cliente**)

FK: cliente referencia CLIENTE (num\_cli) borrado SET NULL

AVERÍA (**num\_parte**, descrip, importe, **pieza**)

FK: pieza referencia PIEZA (num\_pieza) borrado RESTRICT

INCLUYE (**factura**, **avería**)

FK: factura referencia FACTURA (num\_fac) borrado CASCADE

FK: avería referencia AVERÍA (num\_parte) borrado CASCADE

1. Haz el paso a tablas del siguiente esquema entidad-relación, en el que se guarda información sobre una serie de sitos webs.

Si se da de baja a un anunciante, las webs que tuviesen sus anuncios se quedarán sin anunciante.

Si se borra un editor, se borrará automáticamente la web que editara (ERRATA: la relación es (1, 1) (1, n))

Diagrama

Descripción generada automáticamente

ANUNCIANTE (**empresa, num\_agente**, sector)

EDITOR (**CIF**, nombre, teléfono, email)

NOT NULL: email

UNIQUE: email

WEB (**IP**, CMS, **anunciante\_empresa, anunciante\_agente**, tarifa, **editor**)

FK: anunciante\_empresa, anunciante\_agente referencia ANUNCIANTE (empresa, num\_agente) borrado SET NULL

FK: editor referencia EDITOR(CIF) borrado CASCADE

# EJERCICIOS DE E-R EXTENDIDO: DIAGRAMAS

1. Definir la siguiente jerarquía.

Se quiere tener una base de datos de vehículos. Existen dos tipos de vehículos que se tienen en cuenta: vehículos motorizados y vehículos no motorizados.

De los vehículos motorizados interesa saber el número de chasis, el número de motor y el tipo. Los vehículos motorizados pueden ser motos, camiones o automóviles, aunque podría haber otros tipos.

De las motos interesa saber su cilindrada, de los camiones su tara y sus ejes y del automóvil el número de puertas.

En cambio, de los vehículos no motorizados interesa saber el número de ruedas y la categoría. Los vehículos no motorizados podrían ser carretas, triciclos o bicicletas, entre otros. De las carretas interesa saber su tracción, del triciclo su carga y de la bicicleta el número de cambios.

Para cualquier vehículo hay que registrar su código identificativo, color y clase.Diagrama

Descripción generada automáticamente

1. Se desea realizar una base de datos, que contenga información relativa a las carreteras de un país:
   * Las carreteras se encuentran divididas en tramos.
   * Un tramo siempre pertenece a una sola carretera. Un tramo se identifica por un código según su orden: 1º, 2º, 3º…

Todas las carreteras codifican sus tramos igual.

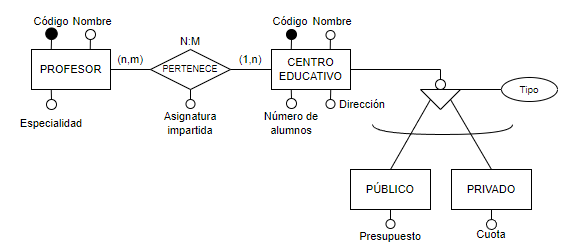
* + Un tramo puede pasar por varios términos municipales, siendo un dato de interés el kilómetro de entrada y el kilómetro de salida.
  + Existen una serie de áreas en las que se agrupan los tramos terminados, cada uno de los cuales no puede pertenecer a más de un área.
  + Los tramos en construcción no tienen área, pero una vez acabados tienen que tenerla.

Establece atributos básicos.

Diagrama

Descripción generada automáticamente

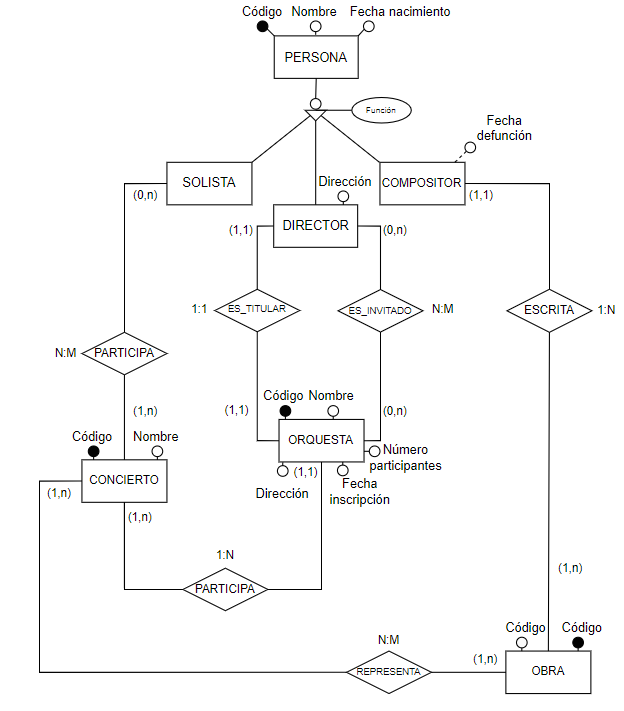
1. Un profesor puede impartir clases en varios centros. Para el profesor interesa conocer su código, nombre, especialidad y la asignatura que imparte en cada centro. Los centros pueden ser públicos o privados. Para cada centro interesa saber su código, nombre, dirección y número de alumnos. Para los centros públicos interesa saber además el presupuesto disponible, y para los privados la cuota.



1. Se desea recoger información sobre el programa de conciertos de música clásica del auditorio nacional, con arreglo a las siguientes directivas.
   * Una orquesta puede interpretar varios conciertos durante la temporada.
   * Como director titular de una orquesta figura una sola persona, pero dicha orquesta puede contar con varios directores invitados. Un director no puede dirigir más de una orquesta.
   * En un mismo concierto pueden interpretarse varias obras.
   * Un mismo solista puede participar en varios conciertos al año, y en un concierto se puede contar con varios solistas.
   * Cada obra es escrita por un solo compositor.

Se desean almacenar los siguientes datos: un código para cada persona que participa en el auditorio, nombre y fecha de nacimiento.

Además: Código de la orquesta, código del concierto, código de la obra, nombre de la orquesta, dirección a la que está adscrita la orquesta, fecha de inscripción de la orquesta, número de componentes que forman la orquesta, dirección del director, nombre que se le ha dado a cada concierto, título de las obras y fecha de defunción del compositor en caso de estar difunto.



1. Definir el diagrama Entidad/Relación del siguiente caso, con la notación Chen.

La coordinadora nacional de Organizaciones No Gubernamentales (ONG`s) desea mantener una base de datos de las asociaciones de este tipo que existen en nuestro país.

Para ello necesita almacenar información sobre cada asociación, sus participantes (socios y trabajadores) y los proyectos que realizan. De las asociaciones se desea almacenar su cif, denominación, dirección y provincia, su tipo de actividad (ecologista, integración, desarrollo...), así como si está declarada de utilidad pública por el Ministerio del Interior.

Cada asociación está formada por socios de los que se precisa conocer su dni, nombre, dirección, provincia, fecha de alta en la asociación, la cuota mensual con que colaboran y la aportación anual que realizan (que se obtendrá multiplicando la cuota mensual por los meses del año).

Los trabajadores de estas organizaciones pueden ser de dos tipos: asalariados y voluntarios.

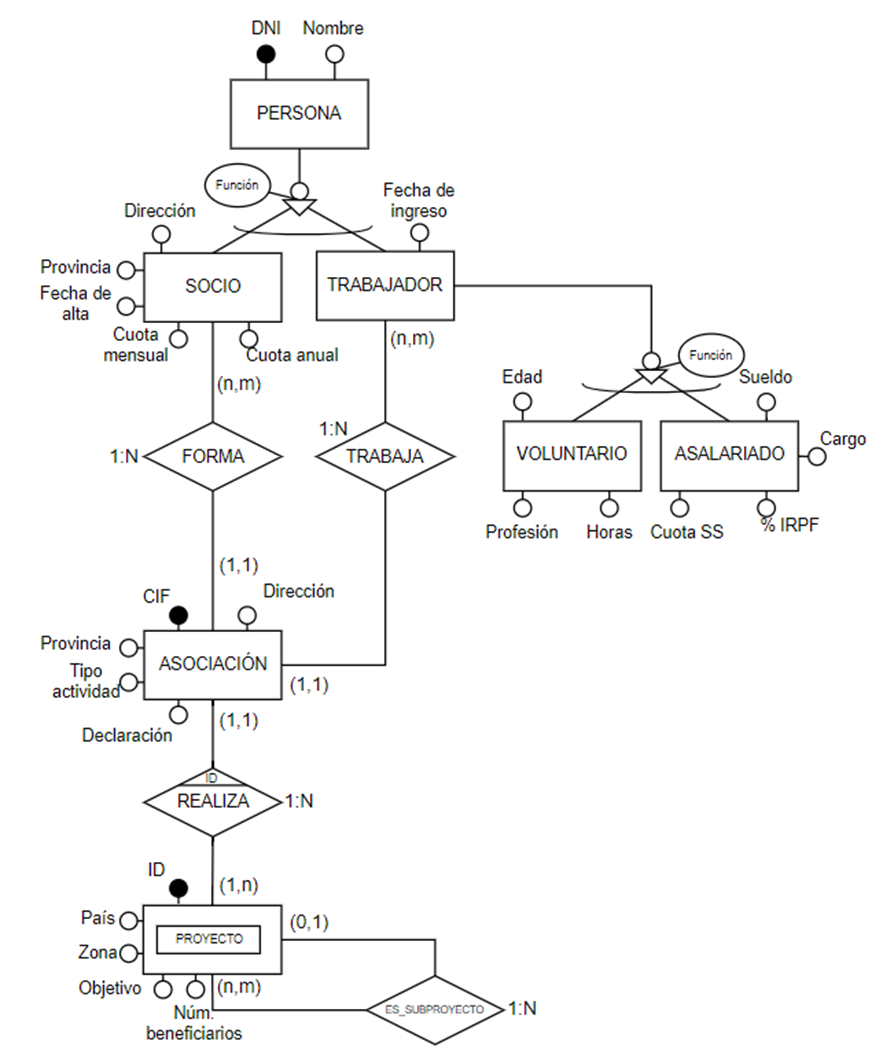
Los asalariados son trabajadores que cobran un sueldo y ocupan cierto cargo en la asociación. Se desea almacenar la cantidad que éstos pagan a la seguridad social y el tanto por ciento de IRPF que se les descuenta.

Los voluntarios trabajan en la organización desinteresadamente, siendo preciso conocer su edad, profesión y las horas que dedican a la asociación a efectos de cálculo de estadísticas.

Cada trabajador se identifica por su dni, tiene un nombre y una fecha de ingreso.

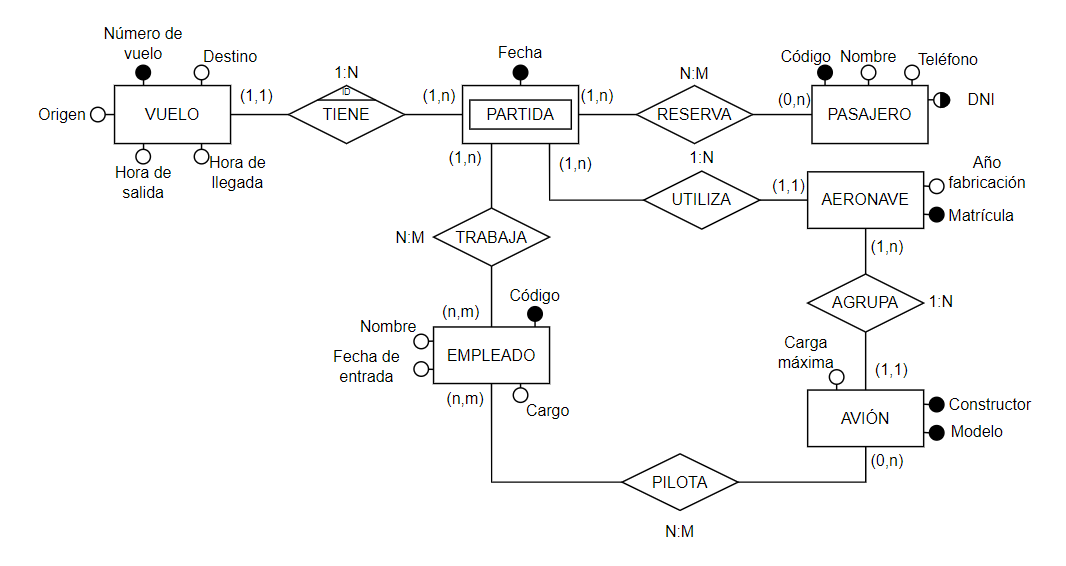
Un socio no puede ser trabajador de la asociación.

Las asociaciones llevan a cabo proyectos. Un proyecto sólo puede pertenecer a una asociación en concreto. De cada proyecto se desea almacenar su número de identificación dentro de la asociación, en qué país se lleva a cabo y en qué zona de éste, así como el objetivo que persigue y el número de beneficiarios a los que afecta. Un proyecto se compone a su vez de subproyectos (que tienen entidad de proyectos).



1. Sea una compañía aérea sobre la que pretendemos diseñar una Base de Datos, tomando en consideración que:
   * Un mismo vuelo puede tener partidas en diferentes fechas (el número de vuelo, destino, origen, hora de salida y hora de llegada son iguales pero en distintas fechas).
   * Los pasajeros reservan plazas para las partidas (considerando que una partida es un número de vuelo y la fecha en que se realiza). A veces las partidas se quedan vacías.
   * Cada aeronave es de un tipo de avión. El tipo de avión se distingue por el constructor y modelo.
   * Algunos empleados (pilotos) son capaces de tripular diversos tipos de aviones.
   * Los empleados de la compañía, pueden viajar como tripulación en distintas partidas. Una tripulación está compuesta por al menos 4 empleados.
   * Cada partida se realiza en un avión concreto (aeronave). Todos los aviones se usan para al menos una partida.

Además de la información anterior, se desean almacenar los siguientes datos: código del pasajero, DNI pasajero, nombre del pasajero, teléfono del pasajero, código de empleado, nombre empleado, fecha de entrada en la empresa de cada empleado, cargo que ocupa en la empresa cada empleado, modelo del avión, carga máxima (depende del tipo de avión), matrícula y año de fabricación de la aeronave.



# EJERCICIOS DE E-R EXTENDIDO: PASO A TABLAS

1. Dado el modelo E/R:

Diagrama

Descripción generada automáticamente

1. Haz el paso a tablas eliminando los subtipos:

PROFESOR (**código**, nombre, especialidad)

CENTRO (**código**, nombre, dirección, num\_alumno, presupuesto, cuota, tipo)

NOT NULL: tipo.

ENSEÑA (**profesor, centro**, asignatura)

FK: profesor referencia PROFESOR (código) borrado CASCADE.

FK: centro referencia CENTRO (código) borrado CASCADE.

¿Cuál es el inconveniente de las tablas generadas por este sistema?

Se generarán nulos en la tabla centro.

1. Haz el paso a tablas eliminando el supertipo:

PROFESOR (**código**, nombre, especialidad)

CENTRO\_PÚBLICO (**código**, nombre, dirección, num\_alumno, presupuesto)

CENTRO\_PRIVADO (**código**, nombre, dirección, num\_alumno, cuota)

ENSEÑA\_PÚBLICO (**profesor, centro**, asignatura)

FK: profesor referencia PROFESOR (código) borrado CASCADE.

FK: centro referencia CENTRO\_PÚBLICO (código) borrado CASCADE.

ENSEÑA\_PRIVADO (**profesor, centro**, asignatura)

FK: profesor referencia PROFESOR (código) borrado CASCADE.

FK: centro referencia CENTRO\_PRIVADO (código) borrado CASCADE.

¿Cuál es el inconveniente de las tablas generadas por este sistema?

Se puede generar duplicidades de datos, por ejemplo, un profesor que trabaje en un centro público y en un centro privado.

1. Haz el paso a tablas eliminando la jerarquía:

PROFESOR (**código**, nombre, especialidad)

CENTRO (**código**, nombre, dirección, num\_alumno, tipo)

ENSEÑA (**profesor, centro**, asignatura)

FK: profesor referencia PROFESOR (código) borrado CASCADE.

FK: centro referencia CENTRO (código) borrado CASCADE.

PÚBLICO (**centro**, presupuesto)

FK: centro referencia CENTRO (código) borrado CASCADE.

PRIVADO (**centro**, cuota)

FK: centro referencia CENTRO (código) borrado CASCADE.

¿Cuál es el inconveniente de las tablas generadas por este sistema?

Se requieren múltiples búsquedas para acceder a los datos de PÚBLICO y PRIVADO, una primera en CENTRO para conseguir el código del centro y otra ya en PÚBLICO o PRIVADO para conseguir los datos propios de cada tabla.

1. Haz el paso a tablas del siguiente modelo sobre una compañía aérea

Diagrama

Descripción generada automáticamente

Ten en cuenta:

1. Si se borra una partida, se borrarán automáticamente las reservas de esa partida.
2. Si se borra un pasajero, se borrarán automáticamente las reservas de ese pasajero.
3. Si se da de baja a un empleado, todas las partidas en las fuera a viajar no se borrarán pero se quedarán con el empleado “suplente”.
4. No puede despedirse a un empleado (dar de baja) si puede tripular un tipo de avión.
5. No puede borrase un tipo de avión si hay aeronaves de ese tipo.
6. Si de da de baja a una aeronave, se anulan automáticamente todas sus partidas.

VUELO (**número vuelo**, origen, destino, hora salida, hora llegada)

PARTIDA (**fecha, vuelo, aeronave**)

FK: vuelo referencia VUELO (número vuelo) borrado CASCADE

FK: aeronave referencia AERONAVE (matrícula) borrado CASCADE

PASAJERO (**código**, dni, tlf, nombre)

NOT NULL: dni

UNIQUE: dni

RESERVA (**partida, vuelo, pasajero**)

FK: partida, vuelo referencia PARTIDA (fecha, vuelo) borrado CASCADE

FK: pasajero referencia PASAJERO (código) borrado CASCADE

EMPLEADO (**código**, nombre, fecha entrada, cargo)

VIAJA (**partida, vuelo, empleado**)

FK: partida, vuelo referencia PARTIDA (fecha, vuelo) borrado CASCADE

FK: empleado referencia EMPLEADO (código) borrado SET DEFAULT “suplente”

TIPO\_AVIÓN (**constructor, modelo**, carga máxima)

TRIPULA (**empleado, avión\_constructor, avión\_modelo**)

FK: empleado referencia EMPLEADO (código) borrado RESTRICT

FK: avión\_constructor, avión\_modelo referencia TIPO\_AVIÓN (constructor, modelo) borrado CASCADE

AERONAVE (**matricula**, año fabricación, **avión\_constructor, avión\_modelo**)

FK: avión\_constructor, avión\_modelo referencia TIPO\_AVIÓN (constructor, modelo) borrado RESTRICT.

# EJERCICIOS DE E-R: NORMALIZACIÓN DE TABLAS

1. Normaliza la tabla hasta 3FN:

Tabla

Descripción generada automáticamente

Dependencia funcional:

* + Cod-pre à Descripción, pvp, tallas

PRENDA (**cod-pre**, descripción, pvp)

TALLA (**prenda, talla**)

FK: prenda referencia PRENDA (cod-pre) borrado CASCADE.

1. Dada la siguiente relación utilizada para almacenar información sobre los artículos que un dependiente vende, además de información del propio dependiente, normalizarla (si no lo está) hasta 3FN.

Descripción de los campos:

* + DNI: dni del dependiente
  + Calle, ciudad y comunidad: dirección del dependiente
  + Cod\_art: código del artículo vendido
  + Cant: cantidad del artículo vendido

Tabla

Descripción generada automáticamente

Nota: identifica en primer lugar las dependencias funcionales.

Supón que el nombre de ciudad no puede repetirse en distintas comunidades autónomas.

Dependencia funcional:

* + DNI à Calle, ciudad.
  + Ciudad à comunidad.
  + DNI, cod-art à cantidad

Tablas:

DEPENDIENTE (**DNI**, calle, **ciudad**)

FK: ciudad referencia CIUDAD\_COMUNIDAD (ciudad) borrado CASCADE.

CIUDAD\_COMUNIDAD (**ciudad**, comunidad)

VENTA (**dependiente, cod-art**, cantidad)

FK: dependiente referencia DEPENDIENTE (DNI) borrado CASCADE.

1. Normalizar hasta 3FN una relación T con atributos A, B, C, D y E, cuyo diagrama de dependencias funcionales es el siguiente:

Una captura de pantalla de un celular con texto e imagen

Descripción generada automáticamente con confianza baja

A à D

B à E

A, B à C

TABLA\_A (**A**, D)

TABLA\_B (**B**, E)

TABLA\_C (**A, B,** C)

FK: A referencia TABLA\_A (A) borrado CASCADE.

FK: B referencia TABLA\_B (B) borrado CASCADE.

1. Se tiene la siguiente tabla referente a las solicitudes de préstamos en un banco, con las siguientes características:
   * Cada préstamo tiene un solicitante (SOLIC) y se identifica por su código (COD-PRÉSTAMO).
   * Cada préstamo tiene un tipo de préstamo (TIPO-PRÉSTAMO).
   * Cada solicitante tiene una dirección (DIR-SOLIC) y un teléfono de contacto (TEL-SOLIC).
   * Cada solicitante puede hacer muchas solicitudes de préstamo.

T (COD-PRESTAMO, SOLIC, TIPO-PRESTAMO, DIR-SOLIC, TEL-SOLIC)

¿Está en 1FN? ¿Está en 2FN? ¿Está en 3FN?

En caso de que no lo esté, normaliza a 3FN.

Nota: identifica en primer lugar las dependencias funcionales.

COD-PRESTAMO à SOLIC, TIPO-PRÉSTAMO

SOLIC à DIR-SOLIC, TEL-SOLIC

PRESTAMO (**COD-PRESTAMO**, **SOLIC**, TIPO-PRÉSTAMO)

FK: SOLIC referencia SOLICITANTE (SOLIC) borrado CASCADE.

SOLICITANTE (**SOLIC**, DIR-SOLIC, TEL-SOLIC)

1. Normalizar hasta 3FN la siguiente tabla T con: NIF, NOMBRE, MAIL, CIUDAD, PAÍS, TELÉFONO y PREFIJO, que presenta una BD con información sobre personas.
   * Una persona solo tiene un teléfono pero puede tener varios emails.
   * El PREFIJO es el prefijo telefónico de cada país (por ejemplo, en España sería +34)
   * Supón que no hay nombres de ciudades iguales en distintos países.

Nota: identifica en primer lugar las dependencias funcionales.

NIF à NOMBRE, TELÉFONO, MAIL, CIUDAD

CIUDAD à PAÍS

PAÍS à PREFIJO

PERSONA (**NIF**, NOMBRE, TELÉFONO, **CIUDAD**)

FK: CIUDAD referencia CIUDAD\_PAIS (CIUDAD) borrado CASCADE.

CORREO (**PERSONA, MAIL**)

FK: PERSONA referencia PERSONA (NIF) borrado CASCADE.

CIUDAD\_PAIS (**CIUDAD**, **PAIS**)

FK: PAIS referencia CIUDAD\_PAIS (PAIS) borrado CASCADE.

PAIS\_PREFIJO (**PAIS**, PREFIJO)

1. Diseñar una BD para informatizar las facturas que una empresa emite a sus clientes. Características:
   * Cada factura es de un cliente y se identifica por el nº de factura (NUM-FACT).
   * En cada factura se pueden incluir varios servicios (SERVICIO) que serán distintos.
   * En una factura determinada hay un coste por cada servicio (COSTE-SERV) realizado a un cliente (ese coste puede ser distinto en cada factura).
   * Cada cliente se identifica por un único nombre (NOM-CLI) y tiene una dirección (DIR-CLI).
   * Los servicios de una factura se han realizado en una fecha determinada (FECHA-SERVICIO) que no tiene por qué ser la misma para todos los servicios.

Se tiene la tabla T (**NUM-FACT**, SERVICIO, COSTE-SERV, NOM-CLI, DIR-CLI, FECHA-SERVICIO)

¿Está en 1FN? ¿Está en 2FN? ¿Está en 3FN? En caso de que no lo esté, normaliza a 3FN.

Nota: identifica en primer lugar las dependencias funcionales.

Está en 2FN pero no en 3FN porque hay dependencias transitivas.

Dependencias funcionales:

NUM-FACT à NOM-CLI

NUM-FACT, SERVICIO à FECHA-SERVICIO, COSTE-SERV

NOM-CLI à DIR-CLI

Tablas:

CLIENTE (**NOM-CLI**, DIR-CLI)

FACTURA (**NUM-FACT, CLIENTE**)

FK: CLIENTE referencia CLIENTE (NOM-CLI) borrado CASCADE.

FACTURA\_SERVICIO (**FACTURA, SERVICIO**, FECHA-SERVICIO, COST-SERV)

FK: FACTURA referencia FACTURA (NUM-FACT) borrado CASCADE.

1. Expresar mediante tablas normalizadas las siguientes dependencias funcionales:

A, B, C → M, N, S

M → N

B, C → O, P, R

O → P

C → Q

Normalizar hasta la tercera forma normal (3FN). No es necesario marcar Foreign keys.

TABLA\_A (**A, B, C,** **M**, S)

FK: B, C referencia TABLA\_B (B, C) borrado CASCADE.

FK: M referencia TABLA\_C (M) borrado CASCADE.

TABLA\_B (**B, C,** **O**, R)

FK: O referencia TABLA\_D (O) borrado CASCADE.

FK: C referencia TABLA\_E (C) borrado CASCADE.

TABLA\_C (**M**, N)

TABLA\_D (**O,** P)

TABLA\_E (**C**, Q)

1. Se tiene una tabla T con los atributos A, B, C, D, E, F, G, H, I, J, K, L y M. Y que cumple lo siguiente:

A, B, C, D → E, F, G, H

F → G

F → H

H → M

B, C → I, J, K

J → K

C → L

Representar las dependencias funcionales mediante un grafo. Normalizar la tabla hasta 3FN. No es necesario marcar Foreign keys.

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

TABLA\_A (**A, B, C, D,** E, **F**)

FK: F referencia TABLA\_B (F) borrado CASCADE.

FK: B, C referencia TABLA\_D (B, C) borrado CASCADE.

TABLA\_B (**F**, G, **H**)

FK: H referencia TABLA\_C (H) borrado CASCADE.

TABLA\_C (**H**, M)

TABLA\_D (**B, C**, I, **J**)

FK: J referencia TABLA\_E (J) borrado CASCADE.

FK: C referencia TABLA\_F (C) borrado CASCADE.

TABLA\_E (**J**, K)

TABLA\_F (**C**, L)

1. Una empresa desea mecanizar la emisión de sus facturas. Cada factura debe contener los siguientes datos:

NUMFAC, FECFAC, CODCLI, NOMCLI, DIRCLI, CIFCLI, NUMAL {CODAR, NOMAR, UNIPD, PREAR} (líneas repetidas, es decir, una factura tiene varios artículos) DESFC, FORPAGFC

Donde:

* + NUMFAC: número de factura
  + FECFAC: fecha de la factura
  + CODCLI: código del cliente
  + NOMCLI: nombre del cliente
  + DIRCLI: dirección del cliente
  + CIFCLI: cif del cliente
  + NUMAL: número de albarán al que corresponde la factura
  + CODAR: código del artículo
  + NOMAR: nombre del artículo
  + UNIPD: unidades pedidas del artículo
  + PREAR: precio del artículo
  + DESFC: descuento de cada factura, que depende exclusivamente de la misma y no del cliente al que se le emite
  + FORPAGFC: forma de pago de cada factura que depende exclusivamente de la misma y no del cliente al que se le emite.

Se debe tener en cuenta que en una misma factura no pueden existir dos pedidos del mismo artículo.

Se pide: Normalizar hasta la tercera forma normal (3FN)

Nota: es aconsejable determinar las dependencias funcionales.

Dependencias funcionales:

NUMFAC à FECFAC, CODCLI, NUMAL, DESCFC, FORPAGFC

NUMFAC, CODAR à UNIPD

CODCLI à NOMCLI, DIRCLI, CIFCLI

CODAR à NOMAR, PREAR

Tablas:

CLIENTE (**CODCLI**, NOMCLI, DIRCLI, CIFCLI)

NOT NULL, UNIQUE: CIFCLI

ARTÍCULO (**CODAR**, NOMAR, PREAR)

FACTURA (**NUMFAC**, **CLIENTE**, FECFAC, NUMAL, DESCFC, FORPAGFC)

NOT NULL, UNIQUE: NUMAL

FK: CLIENTE referencia CLIENTE (CODCLI) borrada CASCADE.

PEDIDO (**FACTURA, ARTICULO,** UNIPD)

FK: FACTURA referencia FACTURA (NUMFAC) borrado CASCADE.

FK: ARTICULO referencia ARTICULO (CODAR) borrado CASCADE.

1. Se diseña una BD para registrar la compra de armas a traficantes por parte de grupos armados.

Cada GRUPO ARMADO tiene un LIDER y una CAUSA por la que lucha.

Hay TRAFICANTES con HISTORIAL DE DELITOS.

Cada TRAFICANTE fabrica un TIPO DE ARMA en una FÁBRICA determinada. En una FÁBRICA se fabrican varios TIPOS DE ARMAS. Estas FÁBRICAS están por todo el mundo y se conoce el PAIS en que se ubican, así como su TAPADERA.

Cada compra de armas la realiza un GRUPO ARMADO a un TRAFICANTE. Una compra se produce en una FECHA y conlleva un NÚMERO DE ARMAS de cada TIPO DE ARMA. Se pueden hacer compras iguales en fechas diferentes.

Se diseña la siguiente BD:

T1 (**FECHA, G-ARMADO, TRAFICANTE, TIPO-ARMA**, NUM-ARMAS)

FK: G-ARMADO referencia T2(G-ARMADO) actualización y borrado en cascada

FK: TRAFICANTE, TIPO-ARMA referencia T3(TRAFICANTE, TIPO-ARMA) act. y borrado en cascada

T2 (**G-ARMADO**, LIDER, CAUSA)

T3 (**TRAFICANTE, TIPO-ARMA**, HISTORIAL-DELITOS, FÁBRICA, PAIS, TAPADERA)

Indica si está normalizada a 1FN, 2FN o 3FN

En caso de no estarlo, normaliza a 3FN.

Está normalizada en 1FN porque la T3 tiene una dependencia funcional parcial en “historial-delitos”, sólo depende de traficante y no de arma.

Dependencias funcionales:

G-ARMADO à LÍDER, CAUSA

TRAFICANTE à HISTORIAL-DELITOS

TRAFICANTE, TIPO-ARMA à FÁBRICA

FÁBRICA à PAÍS, TAPADERA

FECHA à G-ARMADO, TRAFICANTE, NUM-ARMAS

Tablas:

T1 (**G-ARMADO**, LÍDER, CAUSA)

NOT NULL, UNIQUE: LÍDER

T2 (**FÁBRICA**, PAÍS, TAPADERA)

T3 (**TRAFICANTE**, HISTORIAL-DELITOS)

T4 (**TRAFICANTE, TIPO-ARMA**, **FÁBRICA**)

FK: TRAFICANTE referencia T3 (TRAFICANTE) borrado CASCADE.

FK: FÁBRICA referencia T2 (FÁBRICA) borrado CASCADE.

T5 (**FECHA**, **GRUPO, TRAFICANTE, TIPO-ARMA,** NUM-ARMAS)

FK: GRUPO referencia T1 (G-ARMADO) borrado CASCADE.

FK: TRAFICANTE, TIPO-ARMA referencia T4 (TRAFICANTE, TIPO-ARMA) borrado CASCADE.