

## Vicente Villarreal Hinojo

### Ejercicios modelo relacional- Boletín 01

**Ejercicio 1:** Supermercado En un supermercado hay productos organizados en categorías (frutas, ultramarinos, carnes, pescados, etc). Cada producto pertenece a una única categoría y puede haber categorías que todavía no tengan ningún producto asignado, sin embargo, no puede haber productos sin categoría. De las categorías vamos a guardar el nombre, y de los productos el nombre y el precio, así como el número de productos del supermercado.

#### Modelo relacional:

Categoría (#Codigo\_categoria, nombre) siendo  
Codigo\_categoria entero no nulo,  
nombre es cadena no nulo,

PK(Codigo\_Categoria)

Producto (#Codigo\_producto, nombre, precio, numero\_producto) siendo  
Codigo\_producto entero no nulo,  
nombre es cadena no nulo,  
precio es entero no nulo,  
numero\_producto es cadena no nulo,

PK(Codigo\_Producto)

Organizan (#Codigo\_Producto, Codigo\_categoria) siendo  
Codigo\_Producto entero no nulo,  
Codigo\_categoria entero no nulo,

PK(Codigo\_Producto)  
FK(Codigo\_categoria/ Categoría (Codigo\_Categoria))  
FK(Codigo\_Producto / Producto(Codigo\_Producto)))

**Ejercicio 1 versión 2:** La empresa anterior necesita conocer los datos personales de los clientes (nombre, apellidos, dni, dirección y fecha de nacimiento) a los que vende sus productos. Un cliente puede comprar varios productos a la empresa, y un mismo producto puede ser comprado por varios clientes.

**Ejercicio 1 versión 3:** La empresa también necesita saber la fecha en que cada cliente a comprado cada artículo.

Modelo Relacional:

Categoria (#Codigo\_categoria, nombre) siendo  
Codigo\_categoria entero no nulo,  
nombre es cadena no nulo,

PK(Codigo\_Categoria)

Producto (#Codigo\_producto, nombre, precio, numero\_producto) siendo  
Codigo\_producto entero no nulo,  
nombre es cadena no nulo,  
precio es entero no nulo,  
numero\_producto es cadena no nulo,

PK(Codigo\_Producto)

Organizan (#Codigo\_Producto, Codigo\_categoria) siendo  
Codigo\_Producto entero no nulo,  
Codigo\_categoria entero no nulo,

PK(Codigo\_Producto)  
FK(Codigo\_categoria/ Categoria (Codigo\_Categoria))  
FK(Codigo\_Producto / Producto(Codigo\_Producto)))

Cliente (Nombre, #DNI, Direccion, Apellidos, Fecha\_nacimiento) siendo  
Nombre cadena no nulo,  
DNI es cadena no nulo,  
Apellidos es cadena no nulo,  
Fecha\_Nacimiento es fecha no nulo,

PK(DNI)

Comprar (#DNI, #Codigo\_Producto, Fecha\_de\_Compra) siendo  
DNI cadena no nulo,  
Codigo\_Producto entero no nulo,  
fecha\_De\_Compra fecha no nulo

PK(DNI, Codigo\_Producto, Fecha\_de\_Compra)  
FK(Codigo\_Producto / Producto(Codigo\_Producto)))  
FK(DNI / Cliente(DNI)))

**Ejercicio 2:** Autoescuela En una autoescuela hay varios profesores y varios coches de prácticas. Se quiere guardar información respecto a los alumnos que se matriculan (nombre, apellidos y dni), el profesor que se le asigna (dni, nss, nombre y apellidos) y el coche que conducirán (matrícula, marca, modelo). Cada alumno da clase siempre con el mismo profesor y el mismo coche. El alumno siempre va a conducir el mismo coche para habituarse a él y va a dar clase con el mismo profesor, aunque un profesor puede dar clase en varios alumnos con distintos coches.

Profesor (#NSS, nombre, apellidos, DNI ) siendo  
NSS entero no nulo,  
nombre es cadena no nulo,  
apellidos es cadena no nulo,  
DNI es cadena no nulo;

PK( NSS)

Alumno (#DNI, nombre, apellidos, NSS) siendo  
DNI cadena no nulo,  
nombre es cadena no nulo,  
apellidos es cadena no nulo,  
NSS es entero no nulo;

PK(DNI)  
FK(NSS/ Profesor (NSS))

Coches(#Matricula, modelo, marca, DNI) siendo  
Matricula cadena no nulo,  
modelo es cadena no nulo,  
marca es cadena no nulo,  
DNI es cadena no nulo;

PK(Matricula)  
FK(DNI / Alumno (DNI))

## Ejercicio 2 versión 2:

Vamos a cambiar el modelo, y en este caso el dueño de la autoescuela nos dice que en realidad el alumno debería acostumbrarse a todos los coches, así que el alumno puede cambiar de coche y de profesor de un día para otro, y que además nos interesa saber qué coche y qué profesor le ha dado clase cada día.

Profesor (#NSS, nombre, apellidos, DNI ) siendo  
NSS entero no nulo,  
nombre es cadena no nulo,  
apellidos es cadena no nulo,  
DNI es cadena no nulo;

PK( NSS)

Alumno (#DNI, nombre, apellidos) siendo  
DNI cadena no nulo,  
nombre es cadena no nulo,  
apellidos es cadena no nulo;

PK(DNI)

Imparte(#NSS, #Profesor\_que\_imparte, #DNI) siendo

NSS es entero no nulo,  
DNI es cadena no nulo,  
Profesor\_Que\_Imparte es cadena no nulo

PK(#NSS, #DNI #Profesor\_Que\_Imparte)  
FK(NSS / Profesor(NSS))  
FK(DNI/ Alumno (DNI))

Coches(#Matricula, modelo, marca) siendo  
Matricula cadena no nulo,  
modelo es cadena no nulo,  
marca es cadena no nulo;

PK(Matricula)

Practica (#DNI, #Matricula #Coche\_que\_se\_usa) siendo  
DNI cadena no nulo,  
Coche\_que\_Se\_Usa cadena no nulo,  
Matricula cadena no nulo,

PK( DNI, Matricula, Coche\_que\_se\_usa)  
FK ( DNI/ Alumno(DNI))  
FK(Matricula /Coche (Matricula))

**Ejercicio 3:** Empresa Se quiere guardar información sobre una empresa que posee varios departamentos. Cada departamento tiene un nombre y un presupuesto. Cada departamento tiene una serie de empleados que sólo pueden depender de un único departamento. Cada departamento tendrá un jefe del departamento que tendrá que ser un empleado de la empresa. De los empleados necesitamos guardar el nss, dni, nombre, apellido, si está casado o no, y el número de hijos que tiene. Hay departamentos que dependen de otros departamentos, por ejemplo el departamento de ventas de exterior y nacionales dependen del departamento de ventas, aunque cada uno tiene sus propios empleados.

Departamento (#Nombre, presupuesto) siendo  
Nombre cadena no nulo,  
Presupuesto es cadena no nulo,

PK(Nombre)

Depende(#Nombre) siendo

#Nombre es cadena no nulo

PK(Nombre)  
FK(Nombre/Departamento(Nombre))

Empleado (#NSS, dni, casado, nombre, apellido, num\_hijos, Nombre\_dep, es\_jefe)  
NSS es entero no nulo,  
DNI es cadena no nulo,  
Casado es cadena no nulo,  
nombre es cadena no nulo,  
apellido es cadena no nulo,  
num\_hijos es entero no nulo,  
nombre\_dep es cadena no nulo,  
es\_jefe es booleano no nulo,

PK(DNI)  
FK(Nombre\_Dep/Departamento(Nombre))

**Ejercicio 4:** Banco En un banco se tienen varias sucursales. Un cliente puede ir a cualquier sucursal y abrir una cuenta. En realidad, el cliente puede abrir varias cuentas en la misma o distinta sucursal. Una cuenta puede pertenecer a uno o varios clientes. Sobre una determinada cuenta, se pueden realizar varias transacciones. Cada sucursal viene identificada por un número de sucursal, también interesa conocer la ciudad en la que se encuentra ubicada y el activo disponible. Por su parte, cada cuenta tiene asociado un número de cuenta y saldo. Del cliente, interesa almacenar el DNI, el nombre, la dirección y la ciudad en la que reside. Cada transacción que se realiza sobre una cuenta tiene un número (de transacción) diferente. Además, se marca la fecha del día, el tipo de operación que se realiza (Ingreso/Extracción) y la cantidad que se mueve.

Cliente (#DNI, nombre, direccion, ciudad\_residencia) siendo  
DNI es cadena no nulo,  
nombre es cadena no nulo,  
Dirección es cadena no nulo,  
Ciudad\_Residencia es cadena no nulo

PK(DNI)

Sucursal(#num\_sucursal,activo,ubicacion) siendo

#num\_sucursal es entero no nulo,  
activo es cadena no nulo,  
ubicacion es cadena no nulo,

PK(num\_sucursal)

Cuenta (#Numero\_Cuenta, saldo,num\_sucursal ) siendo  
Numero\_Cuenta entero no nulo,  
saldo es entero no nulo,  
num\_sucursal es entero no nulo;

PK(Numero\_Cuenta)  
FK(num\_sucursal/Sucursal(num\_sucursal))

Abrir (#DNI, #Numero\_Cuenta) siendo  
DNI cadena no nulo,  
Numero\_Cuenta es entero no nulo,

PK(DNI, Numero\_Cuenta)  
FK(DNI /Cliente (DNI))  
FK (Numero\_Cuenta /Cuenta(Numero\_Cuenta))

Transaccion((#numero,cantidad,tipo,fech\_transaccion,Numero\_cuenta) siendo  
numero es entero no nulo,

cantidad es entero,  
tipo es cadena,  
fecha\_transaccion es fecha,  
Numero\_cuenta es entero no nulo,

PK(numero)  
FK(Numero\_cuenta/Cuenta(Numero\_cuenta))

**Ejercicio 5:** Academia de inglés Una persona asiste a clases particulares en una academia de inglés. Cada persona se ubica en un grupo en función de su nivel de inglés y nos interesa saber el nombre del grupo, el nivel (que será único) y el horario. Hay un grupo por cada nivel. En la academia, hay varios profesores de nacionalidades distintas. Cada profesor da clase a varios grupos y queremos guardar el nombre, apellido, nacionalidad y nss de cada profesor. Cada grupo está formado por varios alumnos. A cada grupo sólo le imparte clase un profesor, y del alumno guardaremos su nombre, nacionalidad y número de cuenta bancaria.

Profesor(#NSS, Nacionalidad, nombre, apellido) siendo  
NSS es entero no nulo,  
Nacionalidad es cadena no nulo,  
Nombre es cadena no nulo,  
apellido es cadena no nulo,

PK(NSS)

Grupo (#Nivel, nombre, horario, NSS) siendo  
Nivel entero no nulo,  
nombre es cadena no nulo,  
horario es entero no nulo,  
NSS es entero no nulo;

PK(Nivel)  
FK(NSS /Profesor (NSS))

Alumnos (#N°Cuenta, Nacionalidad, Nombre, Nivel) siendo  
N°Cuenta entero no nulo,  
Nacionalidad es cadena no nulo,  
Nombre es cadena no nulo,  
Nivel es cadena no nulo

PK (N°\_Cuenta)  
FK (Nivel /Grupo (Nivel))

### Ejercicio 6: Centro de salud

En un centro de salud, un médico atiende a varios pacientes y cada paciente está asignado a un solo médico. Tanto de los médicos como de los pacientes guardaremos sus datos personales (nombre, apellido, dni, dirección, etc). Cada médico, del que también necesitaremos su número de cuenta corriente pasa consulta en una o varias salas. En cada sala pasan consulta varios médicos, de las salas tendremos que saber su número (que será unico) y la situación (planta 1, planta 2 o planta 3).

Se quiere saber en qué sala pasa consulta cada médico en cada momento.

Medico (#DNI, Apellidos, direccion, num\_cuenta, nombre) siendo

DNI es cadena no nulo,

Apellidos es cadena no nulo,

direccion es cadena no nulo,

num\_cuenta es entero no nulo,

nombre es cadena no nulo

PK(DNI)

Sala(#Numero\_Sala, Situacion) siendo

Numero\_Sala es entero no nulo,

Situacion es cadena no nulo,

PK(Numero\_Sala)

Pasar\_Consulta(#DNI, #Numero\_Sala #Fecha\_Hora) siendo

DNI es cadena no nulo,

Numero\_sala es entero no nulo,

Fecha\_Hora es fecha no nulo;

PK(DNI, Numero\_Sala, Fecha\_Hora)

FK(DNI /Medico (DNI))

FK(Numero\_Sala/Sala (Numero\_Sala))

Paciente (#DNI, Apellidos, direccion, nombre, dni\_medico) siendo

DNI es cadena no nulo,

Apellidos es cadena no nulo,

direccion es cadena no nulo,

nombre es cadena no nulo

DNI\_Medico es cadena no nulo,

PK(DNI)

FK(DNI\_Medico /medico(DNI))



### Ejercicio 7: Liga de futbol

La Liga de Fútbol Profesional (LFP) tiene el proyecto de implementar una base de datos con estadísticas de la temporada.

En dicha BBDD los futbolistas vendrán identificados por su nº de ficha, interesando además su nombre, apellidos, fecha de nacimiento, peso y estatura. Los equipos vienen identificados por su nombre. También se guardan su año de fundación, nombre del presidente, nº de socios y estadio en el que juega. Un futbolista puede militar en equipos distintos a lo largo de su carrera deportiva, pero no simultáneamente. De cada contrato entre jugador y club interesa reflejar fecha de comienzo, duración, ficha anual y cláusula de rescisión.

Los equipos disputan partidos entre sí, de los que se guarda la fecha, el resultado y la jornada a la que corresponden. Cada jugador participa en diferentes partidos (puede que en ninguno), siendo relevante el nº de minutos disputados, los goles anotados (o recibidos si se trata de un portero) y las tarjetas recibidas.

De los árbitros interesa el nº de colegiado, el colegio arbitral al que pertenece, así como su nombre, apellidos y el nº de temporadas en la categoría. Cada partido lo arbitran cuatro colegiados (árbitro principal, auxiliares de banda y cuarto árbitro), siendo de interés saber la función de cada uno de ellos.

#### Partidos

Partidos (#Id\_Partidos, Fecha, Jornada, Resultado) siendo  
ID\_Partidos cadena no nulo,  
Fecha fecha no nulo,  
Resultado entero no nulo

PK(ID\_Partidos)

#### Arbitros

Arbitros (#Numero\_de\_Colegiado, N°\_Temporadas, Colegio\_Arbital, Apellidos, Nombre)  
siendo  
Numero\_de\_colegiados es entero no nulo,  
N°\_Temporadas entero no nulo,  
Colegio\_Arbital cadena  
Apellidos Cadena no nulo,  
Nombre es cadena no nulo

PK(Numero\_de\_Colegiado)

## Arbitrados

Arbitrados (#ID\_Partido, #Numero\_de\_Colegiado, Funcion) siendo

ID\_Partidos cadena no nulo,  
Numero\_de\_Colegiado entero no nulo,  
Funcion es cadena no nulo

PK(ID\_Partido, Numero\_de\_colegiado)  
FK(ID\_Partido /Partido (Id\_Partido))  
FK(Numero\_de\_Colegiado /Arbitro (Numero\_de\_Colegiado))

## Equipos

Equipos (#Nombre, año\_de\_Fundacion, Nombre\_De\_Presidente, Numero\_de\_Socios, Nombre\_del\_Estadio) siendo

Nombre cadena no nulo,  
Año\_Fundacion fecha no nulo,  
Nombre\_de\_presidente cadena no nulo,  
Numero\_de\_Socios entero no nulo,  
Nombre\_del\_estadio cadena

PK(Nombre)

## Juegan

Juegan(#Id\_Partido, Nombre, Local\_S\_N) siendo  
Id\_Partido cadena no nulo,  
Nombre Cadena no nulo,  
Local\_S\_N Booleana no nulo,

PK(Id\_Partido, Nombre, Local\_S\_N)  
FK(ID\_Partido /Partido ((Id\_Partido))  
FK(Nombre / Equipos ((Nombre))

## Futbolistas

Futbolistas (#Nº\_Ficha, Nombre, Apellido, Fecha\_Nacimiento, Peso, Estatura) siendo  
Nº\_Ficha entero no nulo,  
Nombre cadena no nulo,  
Apellido cadena no nulo,  
Fecha\_Nacimiento fecha no nulo,  
Peso entero  
Estatura entero

PK(Nº\_ficha)

Pertenece

Pertenece (Nombre, Nº\_Ficha, Clausula\_resincion, Ficha\_Anual, Duracion, Fecha\_Comienzo) siendo  
Nombre cadena no nulo,  
Nº\_Ficha entero no nulo,  
Clausula\_resincion entero no nulo,  
Ficha\_Anual fecha no nulo,  
Duaracion enetero no nulo,  
Fecha\_Comienzo fecha no nulo.

PK(Nombre, Nº\_Ficha)  
FK(Nombre /Equipo((nombre))  
FK(nº\_Ficha/Futbolista ((Nº\_Ficha))

Disputan

Disputan(#id\_partido, #nºficha, minutos\_jugados, goles\_anoatados/recibidos, tarjetas) siendo  
id\_partido es entero no nulo,  
nºficha es entero no nulo,  
minutos\_jugados es entero,  
goles\_anoatados/recibidos es entero,  
tarjetas es entero,

PK(id\_partido, nº\_ficha)  
FK(id\_partido/Partidos(id\_partido))  
FK(nº\_ficha/Futbolistas(nº\_ficha))

**Ejercicio 8:** Ajedrez en Villatortas El club de Ajedrez de Villatortas ha sido encargado por la Federación Internacional de Ajedrez de la organización de los próximos campeonatos mundiales que se celebrarán en la mencionada localidad. Por este motivo, desea llevar a una base de datos toda la gestión relativa a participantes, alojamientos y partidas. Teniendo en cuenta que: En el campeonato participan jugadores y árbitros; de ambos se requiere conocer el número de asociado, nombre, dirección, teléfono de contacto y campeonatos en los que ha participado (como jugador o como árbitro). De los jugadores se precisa además el nivel de juego en una escala de 1 a 10. Ningún árbitro puede participar como jugador. Los países envían al campeonato un conjunto de jugadores y árbitros, aunque no todos los países envían participantes. Todo jugador y árbitro es enviado por un único país. Un país puede ser representado por otro país. Cada país se identifica por un número correlativo según su orden alfabético e interesa conocer además de su nombre, el número de clubes de ajedrez existentes en el mismo. Cada partida se identifica por un número correlativo (Cod\_P), la juegan dos jugadores y la arbitra un árbitro. Interesa registrar las partidas que juega cada jugador y el color (blancas o negras) con el que juega. Ha de tenerse en cuenta que un árbitro no puede arbitrar a jugadores enviados por el mismo país que le ha enviado a él. Todo participante participa en al menos una partida. Tanto jugadores como árbitros se alojan en uno de los hoteles en los que se desarrollan las partidas, se desea conocer en qué hotel y en qué fechas se ha alojado cada uno de los participantes. Los participantes pueden no permanecer en Villatortas durante todo el campeonato, sino acudir cuando tienen que jugar alguna partida alojándose en el mismo o distinto hotel. De cada hotel, se desea conocer e nombre, la dirección y el número de teléfono. El campeonato se desarrolla a lo largo de una serie de jornadas (año, mes, día) y cada partida tiene lugar en una de las jornadas aunque no tengan lugar partidas todas las jornadas. Cada partida se celebra en una de las salas de las que pueden disponer los hoteles, se desea conocer el número de entradas vendidas en la sala para cada partida. De cada sala, se desea conocer la capacidad y medios de que dispone (radio, televisión, video...) para facilitar la retransmisión de los encuentros. Una sala puede disponer de varios medios distintos. De cada partida se pretende registrar todos los movimientos que la componen, la identificación del movimiento se establece en base a un número de orden dentro de cada partida: para cada movimiento se guardan la jugada (5 posiciones) y un breve comentario realizado por un experto.

PAIS(#numero\_correlativo,nombre,numero\_clubes,numero\_correlativo\_Representa) siendo  
numero\_correlativo es entero no nulo,  
Nombre es cadena no nulo,  
numero\_clubes es entero,  
numero\_correlativo\_Representa es entero no nulo,

PK(numero\_correlativo)

FK(numero\_correlativo\_Representa/PAIS(numero\_correlativo))

PARTICIPANTE(#numero\_asociado,campeonatos,nombre,direccion,telefono,numero\_correlativo) siendo

numero\_asociado es entero no nulo,  
Campeonatos es entero,  
Nombre es cadena no nulo,  
Telefono es cadena no nulo,  
numero\_correlativo es entero no nulo,

PK(numero\_asociado)  
FK(numero\_correlativo/PAIS(numero\_correlativo))

JUGADOR(#numero\_asociado,nivel) siendo

numero\_asociado es entero no nulo,  
Nivel es entero

PK(numero\_asociado)  
FK(numero\_asociado/PARTICIPANTE(numero\_asociado))

ARBITRO(#numero\_asociado) siendo

numero\_asociado es entero no nulo,

PK(numero\_asociado)  
FK(numero\_asociado/PARTICIPANTE(numero\_asociado))

HOTEL(#nombre,direccion,telefono) siendo

Nombre es cadena no nulo,  
Direccion es cadena,  
Telefono es cadena no nulo,

PK(nombre)

ALOJAN(#numero\_asociado,#nombre,fecha\_entrada,fecha\_salida)siendo

numero\_asociado es entero no nulo,  
Nombre es cadena no nulo,  
fecha\_entrada es fecha,

PK(numero\_asociado,nombre)  
FK(numero\_asociado/PARTICIPANTE(numero\_asociado))  
FK(nombre/HOTEL(nombre))

SALA(#id\_sala,#nombre\_hotel,medios,capacidad) siendo

Id\_sala es entero no nulo,  
Nombre es cadena no nulo,  
Medios es cadena,  
Capacidad es real,

PK(id\_sala,nombre)  
FK(nombre/HOTEL(nombre))

PARTIDA(#cod\_p,numero\_asociado\_Arbitra,id\_sala,entradas\_vendidas,numero\_jornada)  
siendo

cod\_p es entero no nulo,  
numero\_asociado\_Arbitra es entero no nulo,  
id\_sala es entero no nulo,  
entradas\_vendidas es entero,  
numero\_jornada es entero no nulo,

PK(cod\_p)  
FK(numero\_asociado\_Arbitra/ARBITRO(numero\_asociado))  
FK(id\_sala/SALA(id\_sala))  
FK(numero\_jornada/JORNADA(numero\_jornada))

JUEGA(#numero\_asociado,#cod\_partida,color\_fichas) siendo

numero\_asociado es entero no nulo,  
cod\_partida es entero no nulo,  
color\_fichas es cadena,

PK(numero\_asociado,cod\_partida)  
FK(numero\_asociado/JUGADOR(numero\_asociado))  
FK(cod\_partida/PARTIDA(cod\_partida))

MOVIMIENTO(#num\_orden,jugada,comentario,cod\_p) siendo

num\_orden es entero no nulo,  
Jugada es entero no nulo,  
Comentario es cadena,  
cod\_p es entero no nulo,

PK(num\_orden),  
FK(cod\_p/PARTIDA(cod\_p))

JORNADA(#numero\_jornada,año,mes,día) siendo

Numero\_jornada es entero no nulo,

Año es entero no nulo,

Mes es entero no nulo,

Día es entero no nulo,

PK(numero\_jornada)

### **Ejercicio 9: Nóminas Oías S.A.**

La empresa Oías S.A. decide informatizar su nómina. Del resultado del análisis realizado, se obtienen las siguientes informaciones:

A cada empleado se le entregan múltiples justificantes de nómina a lo largo de su vida laboral en la empresa y al menos uno mensualmente.

A cada empleado se le asigna un número de matrícula en el momento de su incorporación a la empresa, y éste es el número usado a efectos internos de identificación.

Además, se registran el NIF del empleado, nombre, número de hijos, porcentaje de retención para Hacienda, datos de cuenta corriente en la que se le ingresa el dinero (banco, sucursal y número de cuenta) y departamentos en los que trabaja. Un empleado puede trabajar en varios departamentos y en cada uno de ellos trabajará con una función distinta. Son datos propios de un justificante de nómina el ingreso total percibido por el empleado y el descuento total aplicado. La distinción entre dos justificantes de nómina se hará, además de mediante el número de matrícula del empleado, mediante el ejercicio fiscal y número de mes al que pertenece y con un número de orden en el caso de varios justificantes de nómina recibidos el mismo mes. Cada justificante de nómina consta de varias líneas (al menos una de ingresos) y cada línea se identifica por un número de línea del correspondiente justificante. Una línea puede corresponder a un ingreso o a un descuento. En ambos casos, se recoge la cantidad que corresponde a la línea (en positivo si se trata de un ingreso o en negativo si se trata de un descuento); en el caso de los descuentos, se recoge la base sobre la cual se aplica para el cálculo de éstos. Toda línea de ingreso de un justificante de nómina responde a un único concepto retributivo. En un mismo justificante, puede haber varias líneas que respondan al mismo concepto retributivo. De los conceptos retributivos se mantiene un código y una descripción. De cara a la contabilidad de la empresa, cada línea de un justificante de nómina se imputa al menos a un elemento de coste. Al mismo elemento de coste pueden imputársele varias líneas. Para cada elemento de coste, se recoge un código, una descripción y un saldo. Entre los elementos de coste se establece una jerarquía, en el sentido de que un elemento de coste puede contener a otros elementos de coste, pero un elemento de coste sólo puede estar contenido en, a lo sumo, otro elemento de coste.

Empleado (#Num\_matricula, numero\_hijos, retencion, nombre, nif, cuenta\_corriente) siendo  
Num\_matricula entero no nulo,  
numero\_hijos es entero no nulo,  
retencion es entero no nulo,  
nombre es cadena no nulo,  
nif es cadena no nulo,  
cuenta\_corriente es cadena no nulo

PK(Num\_Matricula)

Departamento(#Codigo\_departamento) siendo  
Código\_departamento entero no nulo

PK(Codigo\_departamento)

Trabaja(#Num\_matricula, #Codigo\_departamento, #Funcion) siendo  
Num\_matricula es entero no nulo,  
Codigo\_departamento es entero no nulo,  
Funcion es cadena no nulo

PK(Num\_matricula, Codigo\_departamento, Funcion)  
FK(Num\_Matricula /Empleado(Num\_Matricula))  
FK(Codigo\_Departamento / departamento (Codigo\_Departamento))

Nomina (#Num\_Matricula, #mes\_que\_pertenece, #ejercicio\_fiscal, #numero\_de\_origen,  
ingreso, descuento) siendo  
Num\_Matricula es entero no nulo,  
mes\_que\_pertenece es cadena no nulo,  
Ejercicio\_fiscal es entero no nulo,  
Numero\_De\_Origen es entero no nulo,  
ingreso es entero no nulo,  
descuento es entero no nulo

PK(Num\_Matricula, mes\_que\_pertenece, ejercicio\_fiscal, numero\_de\_origen)  
FK(Num\_Matricula/ Empleado (Num\_Matricula))

Lineas (#numero\_linea, #id\_nomina) siendo  
numero\_linea es entero no nulo,  
id\_nomina es entero no nulo,

PK(id\_nomina,numero\_linea)  
FK(id\_nomina/NOMINA(mes\_pertenece,ejercicio\_fiscal,numero\_orden,num\_matricula))



Ingreso(#id\_nomina,#numero\_linea,cantidad) siendo  
id\_nomina es entero no nulo,  
numero\_linea es entero no nulo,  
Cantidad es real,

PK(id\_nomina,numero\_linea)  
FK(id\_nomina/Linea(id\_nomina))  
FK(numero\_linea/Linea(numero\_linea))

Descuento(#id\_nomina,#numero\_linea,cantidad,base) siendo  
id\_nomina es entero no nulo,  
numero\_linea es entero no nulo,  
Cantidad es real,  
Base es real,

PK(id\_nomina,numero\_linea)  
FK(id\_nomina/Linea(id\_nomina))  
FK(numero\_linea/Linea(numero\_linea))

Elemento\_Coste(#codigo,descripcion,saldo) siendo  
Código es entero no nulo,  
Descripcion es cadena,  
Saldo es real no nulo,

PK(código)

Imputa(#id\_linea, #Num\_Linea, #Codigo) siendo  
id\_linea cadena no nulo,  
num\_linea es entero no nulo,  
codigo es entero no nulo

PK(id\_linea, Num\_Linea, Codigo)  
FK(id\_linea / Linea (Id\_linea))  
FK(Num\_linea / Linea (Num\_Linea))

Contiene(#codigo,codigo\_contiene) siendo  
Codigo es entero no nulo,  
codigo\_contiene es entero,

PK(codigo)  
FK(codigo/Elemento\_Coste(codigo))  
FK(codigo\_contiene/Elemento\_Coste(codigo))

Concepto\_Retributivo(#codigo,descripcion,id\_nomina,numero\_linea) siendo  
Codigo es entero no nulo,  
Descripcion es cadena,  
id\_nomina es entero, no nulo,

numero\_linea es entero no nulo

PK(codigo)

FK(id\_nomina/Ingreso(id\_nomina))

FK(numero\_linea/Ingreso(id\_nomina))

**Ejercicio 10: Muebles Sarria** Una empresa dedicada a comercializar cocinas desea aumentar su control sobre aquellos elementos que le afectan. Del resultado del análisis que realiza obtiene las siguientes informaciones: Hay una serie de fabricantes de muebles de cocina. De cada fabricante se dispone de un nombre, una dirección y una relación de números de teléfono. Cada uno de ellos fabrica varios muebles de cocina. Un mueble de cocina tiene una determinada línea, un determinado color, unas dimensiones (ancho, alto y largo), y puede tener una de las siguientes categorías excluyentes: mueble alto, mueble bajo, panel y encimera. De los muebles bajos interesa saber la altura sobre el suelo y de las encimeras interesa saber su tipo (mármol o aglomerado). Cada fabricante puede trabajar con varios distribuidores y cada distribuidor trabaja al menos con un fabricante. De un distribuidor se dispone del nombre, dirección y una relación de números de teléfono. Una cocina la componen una serie de muebles de cocina de distinto tipo, cada mueble de cocina sólo podrá formar parte de una única cocina. De una cocina nos interesa saber el número de muebles que la componen, así como cuántos de ellos hay de cada tipo. Cada cocina la puede vender un único distribuidor en una determinada fecha de venta, aunque cada distribuidor puede vender varias cocinas. Un distribuidor puede ceder una cocina a otro, para que éste pueda venderla. Cada cocina la debe montar al menos un montador, y el mismo montador puede montar varias cocinas. De un montador nos interesa su NIF, nombre, dirección, único número de teléfono y el número de cocinas que ha montado. Cada cocina pueden comprarla uno o varios clientes, y el mismo cliente puede comprar varias cocinas. De un cliente nos interesa su NIF, su nombre, dirección y único número de teléfono.

Fabricante (#Nombre, direccion) siendo

Nombre cadena no nulo,

direccion es cadena no nulo,

PK(Nombre)

Distribuidor (#Nombre, direccion) siendo

Nombre es cadena no nulo,

direccion es cadena no nulo,

PK(Nombre)

Trabaja(#Nombre\_Fabricante, #Nombre\_Distribuidor), siendo  
Nombre\_Fabricante cadena no nulo,  
Nombre\_Distribuidor es cadena no nulo,

PK(Nombre\_Fabricante, Nombre\_Distribuidor  
FK(Nombre\_Fabricante/ Fabricante (Nombre))  
FK(Nombre\_Distribuidor / Distribuidor(Nombre))

CEDE(#nombre,#nombre\_cede) siendo  
Nombre es cadena no nulo,  
nombre\_cede es cadena no nulo,

PK(nombre,nombre\_cede)  
FK(nombre\_cede/DISTRIBUIDOR(nombre))  
FK(nombre/DISTRIBUIDOR(nombre))

Telefono (#Numero\_telefono, nombre\_fabricante, nombre\_distribuidor) siendo  
Numero\_telefono es entero no nulo,  
nomnre\_fabricante es cadena no nulo,  
nombre\_distribuidor es cadena no nulo,

PK(Numero\_telefono)  
FK(Nombre\_Fabricante / Fabricante (Nombre))  
FK(Nombre\_Distribuidor / Distribuidor (Nombre))

Cocina(#Cod\_Cocina, numero\_muebles, tipo, nombre, fecha) siendo  
Cod\_Cocina es cadena no nulo,  
Numero\_muebles es entero no nulo,  
tipo es cadena no nulo,  
nombre es cadena no nulo,  
fecha es fecha

PK(Cod\_Cocina)  
FK(Nombre / Distribuidor (Nombre))

Montador(#Nif, nombre, direccion, telefono) siendo  
Nif cadena no nulo,  
nombre es cadena no nulo,  
direccion es cadena no nulo,  
telefono es entero no nulo,

PK(Nif)

Montar (#Cod\_Cocina, #Nif) siendo  
Cod\_Cocina es entero,  
Nif es cadena no nulo,

PK(Cod\_Cocina, Nif)  
FK(Cod\_Cocina, /cocina (cod\_cocina))  
FK(Nif/ Montador (Nif))

Cliente (#Nif, nombre, direccion, telefono) siendo  
Nif es cadena no nulo,  
nombre es cadena no nulo,  
direccion es cadena no nulo,

PK(NIF)

Es\_Comprada(#Nif, #Cod\_Cocina) siendo  
Nif cadena no nulo,  
Cod\_Cocina es entero no nulo,

PK(Nif, Cod\_Cocina)  
FK(Nif /Cliente (Nif))  
FK(Cod\_Cocina /Cocina (Cod\_Cocina))

Mueble(#Linea, color, dimensiones, Cod\_Cocina) siendo  
Linea cadena no nulo  
Color es cadena no nulo,  
dimensiones es entero no nulo,  
Cod\_cocina es cadena no nulo

PK(Linea)  
FK(Cod\_Cocina /Cocina(Cod\_Cocina))

Mueble alto(#linea) siendo  
Linea es cadena no nulo,

PK(linea)  
FK(linea/Mueble(linea))

Mueble bajo (#linea, altura) siendo  
Linea es cadena no nulo,  
Altura es real,

PK(linea)  
FK(linea/Mueble(linea))

Panel(#linea) siendo  
Linea es cadena no nulo,

PK(linea)  
FK(linea/Mueble(linea))

Encimera(#linea, tipo) siendo  
Linea es cadena no nulo,  
Tipo es cadena,

PK(linea)  
FK(linea/Mueble(linea))

**Ejercicio 11: Viajes Olías S.A** Se desea diseñar una base de datos sobre la información de las reservas de una empresa dedicada al alquiler de automóviles teniendo en cuenta que: Un determinado cliente puede tener en un momento dado hechas varias reservas. De cada cliente se desea almacenar su DNI, nombre, dirección y teléfono. Además dos clientes se diferencian por un código único. Cada cliente puede ser avalado por otro cliente de la empresa. Una reserva la realiza un único cliente pero puede involucrar a varios coches. Es importante registrar la fecha de inicio y final de la reserva, el precio del alquiler de cada uno de los coches, los litros de gasolina en el depósito en el momento de realizar la reserva, el precio total de la reserva y un indicador de si el coche o los coches han sido entregados. No se mantienen los datos de reservas anteriores. Todo coche tiene siempre asignado un determinado garaje que no puede cambiar. De cada coche se requiere la matrícula, el modelo, el color y la marca. Cada reserva se realiza en una determinada agencia.

Cliente (#DNI, nombre, direccion, DNI\_Cliente) siendo  
DNI es cadena no nulo,  
Nombre es cadena no nulo,  
direccion es cadena no nulo,  
Dni\_cliente es entero no nulo,

PK(DNI)  
FK(Dni\_cliente / Avala (Dni\_cliente))

Reserva (#Cod\_reserva, agencia, precio\_total\_reserva, DNI, fecha\_inicio, fecha\_fin) siendo  
Cod\_reserva es entero no nulo,  
agencia es cadena no nulo,  
precio\_total\_reserva entero no nulo,  
Dni es cadena no nulo,  
Fecha\_Inicio es fecha no nulo,  
Fecha\_Fin es fecha no nulo,

PK(Cod\_Reserva)

FK(DNI /Cliente (DNI))

Coche(#Matricula, modelo, marca, litro\_gasolina, garaje, precio\_alquiler. Cod\_reserva, entregado) siendo

Matricula es cadena no nulo,

Modelo es cadena no nulo,

Marca es cadena no nulo,

Litro\_gasolina es entero,

garaje es cadena

Precio\_Alquiler es entero no nulo,

Cod\_Reserva es entero no nulo,

entregado es booleano,

PK(Matricula)

FK(Cod\_Reserva/ Reserva (Cod\_Reserva))

**Ejercicio 12: Parques Naturales** La ministra de Medio Ambiente ha decidido crear un sistema de información sobre los parques naturales gestionados por cada comunidad autónoma. Después de realizar un detallado análisis, se ha llegado a las siguientes conclusiones: Una comunidad autónoma (CA) puede tener varios parques naturales. En toda comunidad autónoma existe uno y sólo un organismo responsable de los parques. Un parque puede estar compartido por más de una comunidad. Un parque natural se identifica por un nombre, fue declarado en una fecha, se compone de varias áreas identificadas por un nombre y caracterizadas por una determinada extensión. Por motivos de eficiencia se desea favorecer las consultas referentes al número de parques existentes en cada comunidad y la superficie total declarada parque natural en cada CA. En cada área forzosamente residen especies que pueden ser de tres tipos: vegetales, animales y minerales. Cada especie tiene una denominación científica, una denominación vulgar y un número inventariado de individuos por área. De las especies vegetales se desea saber si tienen floración y en qué periodo se produce ésta; de las animales se desea saber su tipo de alimentación (herbívora, carnívora u omnívora) y sus periodos de celo; de las minerales se desea saber si se trata de cristales o de rocas. Además, interesa registrar qué especies sirven de alimento de cualquier otra especie y que una especie vegetal no se alimenta de ninguna otra especie. Del personal del parque se guarda el DNI, número de seguridad social, nombre, dirección, teléfonos (domicilio, móvil) y sueldo. Se distinguen los siguientes tipos de personal:

- Personal de gestión: registra los datos de los visitantes del parque y están destinados en una entrada del parque (las entradas se identifican por un número).
- Personal de vigilancia: vigila un área determinada del parque que recorre en un vehículo (tipo y matrícula).
- Personal de conservación: mantiene y conserva un área determinada del parque. Cada uno lo realiza en una especialidad determinada (limpieza, caminos...).
- Personal investigador: tiene una titulación que ha de recogerse y pueden realizar (incluso conjuntamente) proyectos de investigación sobre una determinada especie. Un proyecto de

investigación tiene un presupuesto y un período de realización. Un visitante (DNI, nombre, domicilio y profesión) debe alojarse dentro de los alojamientos de que dispone el parque; éstos tienen una capacidad limitada y tienen una determinada categoría. Los alojamientos organizan excursiones al parque, en vehículo o a pie, en determinados días de la semana y a una hora determinada. A estas excursiones puede acudir cualquier visitante del parque.

Comunidad Autonoma(#nombre,parques\_existentes) siendo

Nombre es cadena no nulo,

parques\_existentes es entero no nulo,

PK(nombre)

Parque Natural (#nombre,superficie,fecha\_declaracion) siendo

Nombre es cadena no nulo,

Superficie es entero,

fecha\_declaracion es fecha,

PK(nombre)

Alojamiento (#nombre,#categoria,capacidad) siendo

Nombre es cadena no nulo,

Categoría es cadena no nulo,

Capacidad es entero,

PK(nombre,categoria)

FK(nombre/Parque Natural(nombre))

Excursion(#num\_excursion,medio,hora,dia,categoria,nombre) siendo

num\_excursion es entero no nulo,

Medio es cadena,

Hora es cadena,

Categoría es cadena,

Nombre es cadena no nulo,

PK(num\_excursion)

FK(nombre/Parque Natural(nombre))

Visitante(#DNI,nombre,domicilio,profesion) siendo

DNI es cadena no nulo,

Nombre es cadena no nulo,

Domicilio es cadena,

Profesión es cadena,

PK(DNI)

Se\_Hospedan(#categoria,#nombre,#DNI) siendo

Categoría es cadena no nulo,

Nombre es cadena no nulo,

DNI es cadena no nulo,

PK(categoria,nombre,DNI)

FK(categoria/Alojamiento(categoria))

FK(nombre/Alojamiento(nombre))

FK(DNI/Visitante(DNI))

Tiene(#nombre\_Comunidad\_Autonoma,#Nombre\_Parque\_Natural,organismo) siendo

nombre\_Comunidad\_Autonoma es cadena no nulo,

nombre\_Parque\_Natural es cadena no nulo,

PK(nombre\_Comunidad\_Autonoma,nombre\_ParqueNatural)

FK(nombre\_Comunidad\_Autonoma/Comunidad\_Autonoma(nombre))

FK(nombre\_Parque\_Natural/Parque\_Natural(nombre))

Personal(#DNI,nss,nombre,direccion,sueldo,nombre\_Parque\_Natural) siendo

DNI es cadena no nulo,

nss es cadena no nulo,

Nombre es cadena no nulo,

Direccion es cadena,

nombre\_Parque\_Natural es cadena no nulo,

PK(DNI)

FK(nombreParqueNatural/Parques\_Naturales(nombre))

Entrada(#numero) siendo

Numero es entero no nulo,

PK(numero)

Personal Gestion(#DNI,numero) siendo

DNI es cadena no nulo,

Numero es entero no nulo,

PK(DNI)

FK(numero/Entrada))

Personal\_Vigilancia(#DNI) siendo

DNI es cadena no nulo,

PK(DNI)

Vehículo (#matricula,tipo) siendo

Matricula es cadena no nulo,

Tipo es cadena no nulo,

PK(matricula)



Personal\_Conservacion (#DNI,area,especialidad) siendo  
DNI es cadena no nulo,  
Area es cadena,  
Especialidad es cadena no nulo,

PK(DNI)

Proyecto\_Investigacion(#cod\_proyecto,duracion,presupuesto) siendo  
cod\_proyecto es entero no nulo,  
Duración es entero,  
Presupuesto es real,

PK(cod\_proyecto)

Proyecto\_Investigador(#DNI,titulacion) siendo  
DNI es cadena no nulo,  
Titulación es cadena no nulo,

PK(DNI)

Realiza(#cod\_proyecto,#DNI) siendo  
cod\_proyecto es entero no nulo,  
DNI es cadena no nulo,

PK(cod\_proyecto,DNI),  
FK(cod\_proyecto/Proyecto\_InvestigacionN(cod\_proyecto)),  
FK(DNI/Personal(DNI))

Movil(#num\_tlf,dni) siendo  
num\_tlf es cadena no nulo,

PK(num\_tlf),  
FK(dni/PersonalL(dni))

Areas(#nombre\_area,extension,nombre\_Parque\_Natural) siendo  
Nombre\_area es cadena no nulo,  
Extension es entero no nulo,  
nombre\_Comunidad\_Autonoma es cadena no nulo,

PK(nombre\_area)  
FK(nombre\_Parque\_Natural/Parque\_Natural(nombre))

Especie(#cod\_especie,denominacion\_vulgar,num\_inventariado,denominacion\_cientifica)  
siendo  
cod\_especie es entero no nulo,  
denominacion\_vulgar es cadena no nulo,  
num\_inventariado es entero,

denominacion\_cientifica es cadena,

PK(cod\_especie)

Alimenta(#cod\_especie,#cod\_especie\_Alimenta) siendo  
cod\_especie es entero no nulo,  
cod\_Especie\_Alimenta es entero,

PK(cod\_especie,cod\_especie\_Alimenta)  
FK(cod\_especie/Especie(cod\_especie))  
FK(cod\_especieAlimenta/Especie(cod\_especie))

Vegetal(#cod\_especie,floracion,periodo) siendo  
Cod\_especie es entero no nulo,  
Floración es cadena,  
Periodo es cadena,

PK(cod\_especie)  
FK(cod\_especie/Especie(cod\_especie))

Animal(#cod\_especie,tipo\_alimentacion) siendo  
cod\_especie es entero no nulo,  
Alimentación es cadena,

PK(cod\_especie)  
FK(cod\_especie/Especie(cod\_especie))

Mineral(#cod\_especie,cristal,roca) siendo  
cod\_especie es entero no nulo,  
Cristal es logico,  
Roca es logico,

PK(cod\_especie)  
FK(cod\_especie/Especie(cod\_especie))

## Ejercicio Agencia Modelo Relacional - Vicente Villarreal Hinojo

**Titular** (#Cod\_Empleado, nombre, telefono, fecha\_nacimiento, titulacion, zona) siendo  
Cod\_Empleado cadena no nulo,  
nombre es cadena no nulo,  
telefono es entero no nulo,  
fecha\_Nacimiento es fecha no nulo,  
titulacion es cadena no nulo  
zona es cadena no nulo

PK(Cod\_Empleado)  
FK(Zona /Agencia (zona))

**Vendedor**(#Cod\_Empleado, nombre, telefono, fecha\_nacimiento, porcentaje, zona) siendo  
Cod\_Empleado cadena no nulo,  
nombre es cadena no nulo,  
telefono es entero no nulo,  
fecha\_Nacimiento es fecha no nulo,  
porcentaje es entero no nulo

PK(Cod\_empleado)  
FK(Zona, /Agencia (zona))

**Agencia** (#zona, direccion, fax ) siendo  
zona es cadena no nulo,  
direccion es cadena no nulo,  
fax es cadena no nulo

PK(Zona)

**Telefono** (#Num\_Telefono, Zona)siendo  
Num\_Telefono es entero no nulo  
Zona es cadena no nulo

PK(Num\_telefono)  
FK(Zona /Agencia (Zona))

**Inmueble** (#Codigo, propietario, direccion, superficie,Zona, hipotecado, precio\_venta, precio\_alquiler, fianza) siendo  
Codigo es cadena no nulo,  
propietario es cadena no nulo,  
direccion es cadena no nulo,  
superficie es entero no nulo  
Zona es booleano no nulo,  
hipotecado es cadena no nulo,

precio\_venta es entero no nulo,  
precio\_alquiler es entero no nulo,  
fianza es entero no nulo

PK(Codigo)  
FK(Zona /Agencia (zona))

**Local** (#Codigo\_local , licencia) siendo  
Codigo\_local es cadena no nulo,  
licencia es booleano no nulo,

PK(Codigo\_local es entero no nulo)  
FK(Código\_local /Inmueble (Codigo))

**Piso** (#Codigo\_piso, direccion, num\_habitaciones, num\_baños) siendo  
Codigo\_piso cadena no nulo,  
num\_habitaciones entero no nulo,  
num\_baños entero no nulo

PK(Codigo\_piso es entero no nulo)  
FK(Código\_piso /Inmueble (Codigo))