

Modelado Conceptual: Exemplos e exercicios

Laboratorio de Bases de Datos
Universidade da Coruña

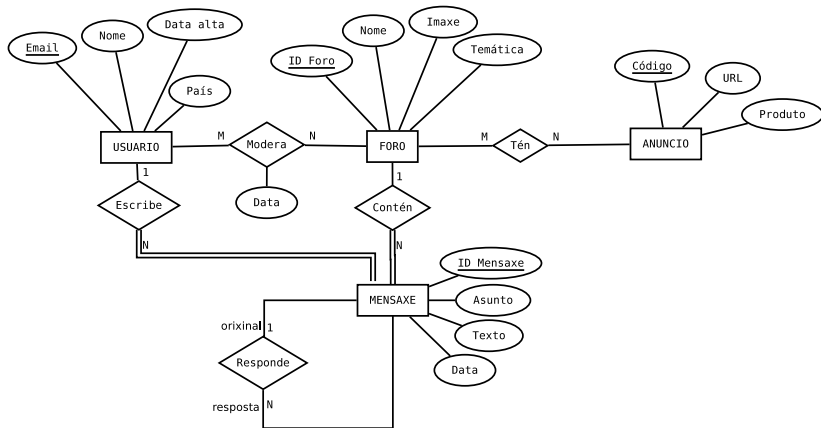


Curso 2015-2016

Exercicio: Foro de mensaxes

- ▶ Queremos dar soporte a unha web de foros de mensaxes cunha pequena base de datos. Nesta web temos usuarios rexistrados, identificados por un email, e dos que se sabe un nome, a data de alta, e o país de orixe.
- ▶ Existe unha serie de foros para as distintas temáticas. De cada foro temos un nome, unha imaxe, e a temática. Cada foro identificarase mediante in “ID” de foro.
- ▶ Algúns dos foros son moderados por usuarios. Rexistraremos que usuarios moderan cada foro, e desde cando. Un usuario pode moderar varios foros.
- ▶ Os usuarios escriben mensaxes dentro dun foro. Cada mensaxe terá un “ID” de mensaxe, o asunto, e un texto. A base de datos rexistrará tamén o usuario autor, e a data e hora de escritura da mesma.
- ▶ As mensaxes poden ser resposta a outras mensaxes (do mesmo foro).
- ▶ Finalmente, existen anuncios (dos que se sabe un código único, o nome do produto, e a URL onde está a páxina ou frame HTML co propio anuncio). Os anuncios estarán asociados ós foros onde o produto estea relacionado coa temática, polo que o mesmo anuncio pode estar en varios foros (e un foro pode ter varios anuncios).

Exercicio: Foro de mensaxes — Solución



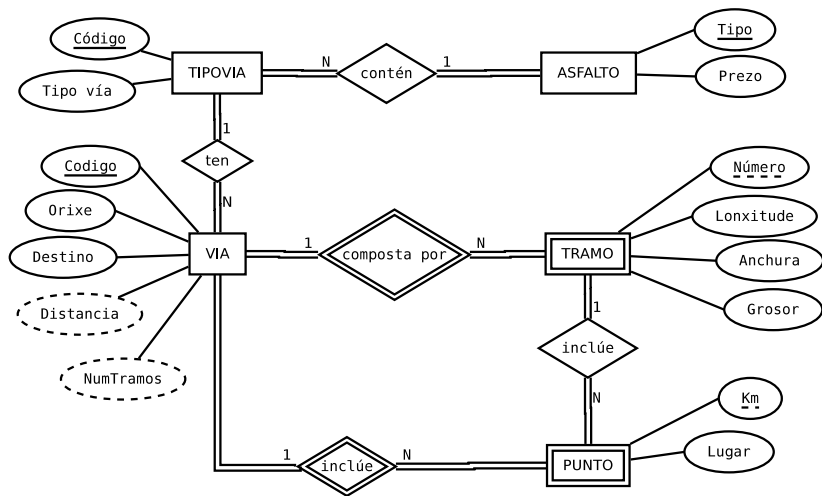
Exercicio: Foro – Esquema Relacional

- ▶ USUARIO(Email, Nomeus, Pais, DataAlta)
- ▶ FORO(IDF, NomeF, Imaxe, Tematica)
- ▶ MENSAXE(IDM, Asunto, Texto, Email, IDF, IDMOrix)
- ▶ ANUNCIO(Cod, URL, Produto)
- ▶ MODERA(Email, IDF, DataM)
- ▶ FORO_ANUNCIA(IDF, Cod)

Exercicio: Rede viaria

- ▶ Queremos xestionar a rede de vías de comunicacións terrestre, considerando todos os tipos de vías: autoestradas, estradas nacionais, comarcais, locais, etc.
- ▶ Cada estrada será dun tipo, estará identificada por un código, e percorrera unha distancia en km entre dous lugares.
- ▶ Dividimos cada estrada en tramos de determinada lonxitude, cunha anchura determinada, que se identifican por número de tramo dentro de cada estrada. Queremos saber o número de tramos de cada estrada.
- ▶ Temos anotados determinados puntos kilométricos, que pertencen a un tramo, identificados polo kilómetro a partir do inicio da estrada, indicando o lugar onde está situado cada un.
- ▶ Temos tipificados diversos tipos de asfalto, de diversos prezos. Para simplificar, asociamos un tipo de asfalto a cada tipo de estrada. Ademais indicaremos, en cada tramo, o grosor dese asfalto.

Exercicio: Rede viaria — Solución



Exercicio: Rede viaria – Modelo Relacional

- ▶ ASFALTO(TiposAsf, Preço)
- ▶ TIPOVIA(CodTipoVia, NomeTipoVia, TipoAsf)
- ▶ VIA(CodVia, Orixe, Destino, CodTipoVia, Tramos)
- ▶ TRAMO(CodVia, NumTramo, Lonxitude, Anchura, Grosor)
- ▶ PUNTO(CodVia, Km, Lugar, NumTramo)

Exercicio: Obras

Queremos deseñar unha base de datos para unha pequena empresa de construción.

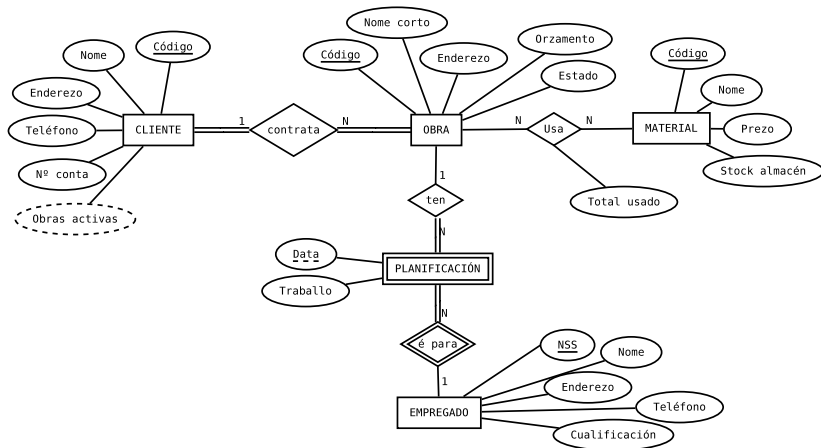
Esta empresa conta con empregados, identificados polo seu NSS e dos que sabemos o seu nome, enderezo, teléfono e a súa cualificación (peón, oficial, etc.). A empresa levará a cabo traballos (obras) para unha serie de clientes. Cada cliente ten un código único asociado, un nome, un enderezo, un teléfono e un número de conta.

Un cliente pode ter unha ou varias obras (non gardamos datos de clientes para os que non temos obras), e unha obra será só dun cliente. Cada obra identifícase por un código único, terá un nome corto, un enderezo, un orzamento, e o estado da obra (sen comezar, activa, rematada). Tamén queremos que o modelo reflicta cantas obras activas temos de cada cliente.

Temos tamén un inventario de materiais, de xeito que cada material se identifica por un código, ten un nome, un prezo, e a cantidade que temos del no almacén. Queremos saber a cantidade total de cada material que se usou en cada obra.

Finalmente, esta base de datos vai servir tamén para facer unha planificación por días do traballo de cada empregado. Así, queremos saber a obra na que estivo cada un dos nosos empregados en cada día, e o traballo que fixo (encofrar, recebar, poñer ladrillo, etc.). Supoñemos que un empregado só está nunha sóa obra durante cada día.

Exercicio: Obras — Solución



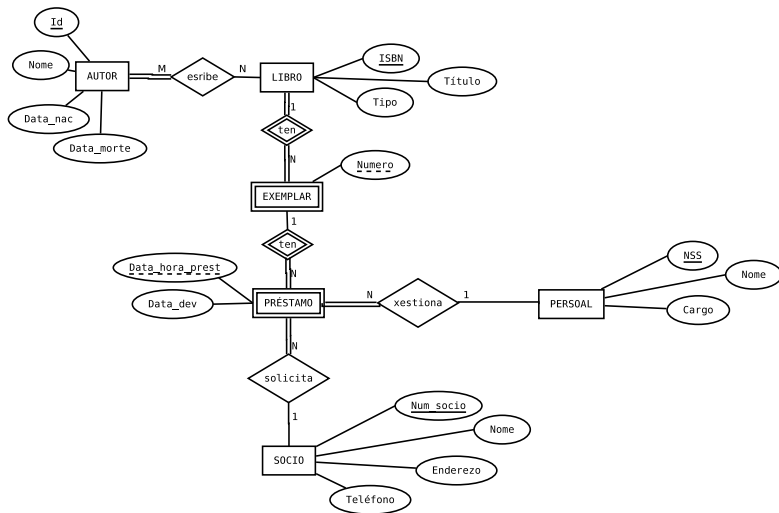
Exercicio: Obras – Modelo Relacional

- ▶ CLIENTE (CODCLI, NOMECLI, ENDEREZOCLI, TELEFCLI, CONTA)
- ▶ OBRA(CODOB, NOME_CORTO, ENDEREZOOB, ORZAMENTO, ESTADO, CODCLI)
- ▶ USA(CODOB, CODMAT, TOTAL)
- ▶ MATERIAL(CODMAT, NOMEMAT, PREZO, STOCK)
- ▶ EMPREGADO(NSS, NOMEEMP, ENDEREZOEMP, TELEFEMP, CUALIF)
- ▶ PLANIFICACION(NSS, DATA, CODOB, TRABAJO)

Exemplo: Biblioteca

- ▶ Na biblioteca temos libros, dos que sabemos o seu ISBN, título e autores que o escribiron.
- ▶ Dos autores dos que temos libros, sabemos o seu nome, a data de nacemento, e a de morte se xa non están vivos.
- ▶ Hai libros de distintos tipos: novela, ensaio, poesía, etc.
- ▶ Na biblioteca hai un determinado número de exemplares de cada libro, identificados polo ISBN do libro seguido dun número correlativo.
- ▶ Do do persoal da biblioteca sabemos o NSS, o nome, e o cargo (dirección ou administración).
- ▶ Para poder tomar prestado un libro hai que ser socio da biblioteca. Cada socio identifícase por un número, e ademais saberemos o nome, o seu enderezo, e un teléfono de contacto.
- ▶ Queremos levar un control dos préstamos, sabendo quen foi o socio que levou un exemplar, e que empregado o xestionou. Cada préstamo identifícase polo exemplar seguido da data e hora actual. Tamén queremos saber cando se devolve o libro.

Exemplo: Biblioteca— Solución



Exemplo: Biblioteca – Modelo Relacional

- ▶ AUTOR(IdAutor, NomeAutor, Datanac, Datamorte)
- ▶ LIBRO(ISBN, Titulo, Tipo)
- ▶ ESCRIBE(IdAutor, ISBN)
- ▶ EXEMPLAR(ISBN, Numero)
- ▶ PERSOAL(NSS, NomePers, Cargo)
- ▶ SOCIO(NumSocio, NomeSocio, Enderezo, Telefono)
- ▶ PRESTAMO(ISBN, Numero, Datahoraprest, Datadev, NSS, NumSocio)

Exercicio: Apps

Queremos deseñar unha base de datos para dar soporte a un sistema de venda de aplicacións para dispositivos móbiles (tipo Google Play ou App Store).

En primeiro lugar, almacenamos datos sobre os desenvolvedores de software. A cada un asígnaselle un código identificador, e almacenamos o seu nome, enderezo postal, enderezo electrónico, e un numero de conta onde facerlle os pagos.

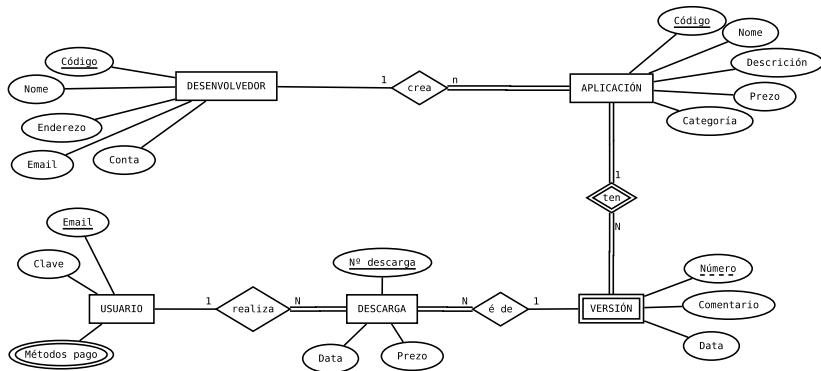
Os desenvolvedores implementan aplicacións (por simplicidade, consideramos que unha aplicación e dun único desenvolvedor). Cada unha terá un código para identificala, un nome, unha descrición, o seu prezo (0 se e gratuíta), e unha clasificación (Xogos, Deportes, Comunicación, Ferramentas, etc.).

De cada aplicación pode haber mais dunha versión. De cada versión gardamos o seu numero, un comentario (para almacenar, por exemplo, os cambios con respecto á versión anterior), e a data desde a que esta dispoñible.

So usuarios autorizados poden realizar descarga/compra de aplicacións. Estes identifícanse polo seu enderezo de correo electrónico, terán unha clave de acceso, e un ou varios métodos preferidos de pago (transferencia bancaria, Paypal, etc.).

Queremos gardar tamén as descargas que os usuarios fan de cada versión dunha aplicación, indicando a data na que se realizou e o que o usuario pagou por ela.

Exercicio: Apps — Solución



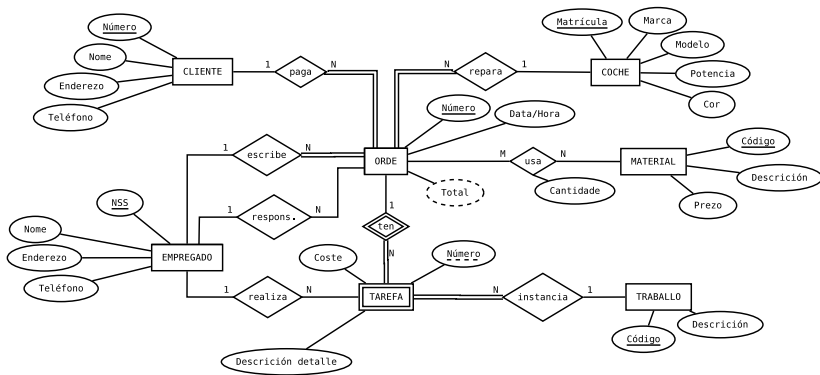
Exercicio: Apps – Modelo Relacional

- ▶ DESENVOLVEDOR (CODD, NOMED, ENDEREZO, EMAILD, CONTA)
- ▶ APLICACION(CODAP, NOMEAP, DESCRICION, PREZOAP, CATEGORIA, CODD)
- ▶ VERSION(CODAP, NUMVERS, COMENTARIO, DATAVERS)
- ▶ USUARIO(EMAILU, CLAVEU)
- ▶ METODO_PAGO(EMAILU, METODO)
- ▶ DESCARGA(NUMDESC, DATADESC, PREZODESC, EMAILU, CODAP, NUMVERS)

Exercicio: Taller

- ▶ Cando unha persoa ven cun coche para arreglar, verificamos se xa é cliente noso, e se non o é dámolo de alta, asignándolle un número de cliente e gardando o nome completo, enderezo de contacto, e número de teléfono. Do coche que o cliente trae a arreglar sabemos a matrícula, a marca, modelo, potencia, e cor.
- ▶ Para realizar calquera reparación, o taller confecciona unha “orde de traballo”, que se codifica cun número, terá a data e hora na que se confecciona, e indicará o cliente que o solicita e para que coche. Ademais deben almacenarse todas as tarefas de reparación que se lle van realizar ó coche (cambio de aceite, reparación de chapa, etc.) nesa orde de traballo, e o coste que estas tarefas supoñen. Unha vez sumados estes costes, aplicándolles o 21 % de IVE, obtemos o importe total da orde de traballo.
- ▶ Os traballos que se realizan no taller están codificados e terán unha descrición xenérica como “reparación de chapa”, para que sexa máis fácil confeccionar a orde de traballo, pero na orde debe incluírse unha descrición a maiores para cada tarefa de reparación que corresponde a un deses traballos (por exemplo: reparación de chapa na aleta dianteira esquerda).
- ▶ Debe incluírse tamén o material (indicando a cantidade) que se utilizou (no caso de que se utilizase algún), xa que debe ser pagado polo cliente. Este material (rodas, aceite, etc.) estará inventariado no taller indicando un código de material, un nome e un prezo.
- ▶ Por outro lado, para depurar responsabilidades, temos tamén información dos empregados do taller (identificados polo NSS e dos que sabemos o nome completo, enderezo e teléfono). Cada orde de traballo será confeccionada (escrita) por un empregado, e designarase un empregado responsable global para esa orde (que non ten que coincidir co que a confeccionou), pero cada tarefa de reparación desa orde pode ser levada a cabo por un empregado diferente, e queremos saber que empregado realiza cada tarefa (supoñemos que só se necesita un empregado para unha tarefa).

Exercicio: Taller — Solución



Exercicio: Taller – Modelo Relacional

- ▶ CLIENTE (NUMC, NOME, ENDEREZO, TELEF)
- ▶ COCHE(MATRICULA, MARCA, MODELO, POTENCIA, COR)
- ▶ EMPREGADO(NSS, NOME, ENDEREZO, TELEF)
- ▶ TRABAJO(CODTRAB, DESCRTRAB)
- ▶ MATERIAL(CODMAT, NOMEMAT, PREZO)
- ▶ ORDE (NUMORD, DATAHORA, NUMC, MATRICULA, NSSE, NSSR)
- ▶ TAREFA(NUMORD, NUMTA, DESCRDETALLE, CODTRAB, NSS)
- ▶ USADO(CODMAT, NUMORD, CANTIDADE)

Exercicio:

O INE (Instituto Nacional de Estadística) garda a seguinte información:

De cada persoa empadronada almacena o seu DNI ou NIE (Número de identidade de estranxeiro), nome, data de nacemento e nivel de estudos. Cada persoa identifícase cun código único.

Gárdase información das vivendas. Cada unha ten unha referencia catastral (única entre todas as referencias catastrais), enderezo e valor fiscal.

Gárdase información tamén de concellos. Cada concello ten un código único, un nome, unha provincia e un número de habitantes.

Cada persoa está empadronada nunha única vivenda, pero pode ser propietaria de cero ou moitas vivendas. Á súa vez, nunha vivenda pode non vivir ninguén, ou vivir varias persoas, pero debe ter polo menos un propietario (aínda que pode ser propiedade de varias persoas). Finalmente, a vivenda está ubicada nun concello.

Debemos saber:

- ▶ Onde vive cada persoa (vivenda e concello).
- ▶ Quen é o propietario (ou propietarios) de cada vivenda, e as vivendas en propiedade de cada persoa.
- ▶ Que vivendas ten cada concello, e que persoas están empadronadas nel.

Exercicio: Armadora de buques

Queremos deseñar a base de datos dunha armadora de buques conxeladores de gran tonelaxe. A base de datos debe conter información dos buques: nome, matrícula, eslora, tonelaxe, data de construción, potencia e porto base. Tamén debe almacenar información dos empregados: número de seguridade social, DNI, nome, dirección, teléfonos e categoría profesional.

Os buques fan campañas pesqueiras de meses de duración fora do porto base. Debemos almacenar información de cada unha destas campañas. De cada campaña almacenamos un código único, o buque que a realiza, a data de saída e a de chegada, o porto de saída e o de chegada (o porto de chegada non ten porque ser igual ao de saída) e a tripulación que leva o buque (que son empregados da compañía).

De cada porto almacenamos un código identificador, a súa cidade, país, calado mínimo e custe diario por amarrar o barco. Obsévese que temos información de portos que non son base de ningún dos nosos barcos, almacenamos información deles porque os barcos poden facer descargas ou saídas/chegadas a eses portos.

Dos empregados que forman parte da tripulación nunha campaña determinada temos que saber que posto desenvolveu/desenvolverá na campaña (capitán, patrón, xefe de mecánicos, etc.). Evidentemente un empregado pode ir enrolado en moitas campañas ao longo do tempo, e temos que ter información de todos eses enroles. Ademais, a pesares de ter unha categoría profesional a día de hoxe, un empregado puido desenvolver distintos postos de traballo en cada campaña na que participou (por exemplo, nunha campaña pode ir de mecánico de 2ª, e na seguinte pode ir como xefe de mecánicos).

En cada campaña, o barco fai unha ou varias descargas de peixe. De cada descarga almacenamos a data e porto (nunha data concreta dunha campaña dun buque, so se pode facer unha descarga como máximo).

Obsévese que non hai ningunha asignación permanente de empregados a barcos.

Exercicio: Xestión de proxectos

Queremos deseñar a BD para un xestor de incidencias (similar a Jira, Trac, Redmine, etc.). Almacenaremos información de cada un dos proxectos que desenvolvemos, que terán un código, un nome, unha descrición e as datas de inicio e fin.

Existirá un conxunto de usuarios que traballan nestes proxectos, identificados por un login, e dos que se almacena a contraseña, o seu nome, email, e o tipo de traballo que realiza (analista, programador, persoal de sistemas, etc.). Un usuario pode traballar en máis dun proxecto á vez, e nun proxecto poden traballar varios usuarios. Queremos almacenar o rol de cada usuario en cada proxecto (xefe de proxecto, informador, desenvolvedor), xa que un mesmo usuario pode realizar distintas tarefas en distintos proxectos.

Cando se produce un erro, ou unha nova necesidade relacionada cun proxecto, un usuario dará de alta unha nova incidencia. Esta incidencia estará identificada por un número único, estará asociada a un proxecto, e tamén saberemos que usuario a deu de alta, e cando. Ademais, terá un título e un texto onde se especifica o que hai que facer. Tamén se indicará o tipo de incidencia (erro ou tarefa de mantemento) e unha prioridade para a mesma (baixa, media, alta, ou urxente), e un estado (o inicial será "Nova"). Tamén é posible asignar a esta incidencia a un usuario para que a xestione (normalmente, un usuario diferente do que a creou).

A medida que a tarefa se xestiona, íranse almacenando actualizacións desta incidencia. Cada unha destas actualizacións terá un número, que será diferente para todas as actualizacións dunha mesma incidencia. Terá un pequeno texto explicando o que se realizou, e o usuario que a levou a cabo. Ademais, unha actualización pode considerarse "resposta" a outra, e queremos saber esta relación entre actualizacións.

Ex: para o proxecto P01 (a web corporativa da empresa), o usuario x.perez deu de alta unha nova incidencia, de mantemento, de título "Desplegar nova versión da web", e asignou a a.glez para a súa realización. a.glez da de alta unha actualización indicando que non ten acceso ós arquivos necesarios para actualizar a web. x.perez responde a esta actualización con outra especificando onde se atopan estes arquivos. Finalmente, a.glez realiza a tarefa e crea unha nova actualización indicando que a incidencia está resolta.

Esta base de datos permitiranos saber o número de incidencias non resoltas que temos para cada un dos proxectos.

Por outro lado, para cada proxecto podemos ter varias versións (test, produción, etc.). Cada versión terá un número diferente do das demais versións dese proxecto, unha descrición, e a URL ou URLs dos repositorios SVN onde se almacena o código fonte desa versión do proxecto. Unha incidencia pode estar relacionada con algunha versión concreta do proxecto.

Temos tamén rexistrados os recursos software, identificándoos con un código, e dos que sabemos o seu nome, versión, e tipo (antivirus, SXBD, etc.). Se nunha actualización dunha incidencia se usou algún destes recursos, queremos sabelo.

Exercicio: TIA

A famosa axencia T.I.A. (Técnicos Investigación Aeroterráquea) onde traballan Mortadelo e Filemón quere modernizar a súa base de datos, para mellorar a planificación e evitar desastres como os últimos que está sufrindo. Encárganos o deseño conceptual e o paso a relacional da base de datos, en base á seguinte información:

A T.I.A. conta cunha serie de axentes, ós que asigna un número de axente único, e dos que sabe tamén o seu nome real, os posibles alias que usa, e as súas habilidades especiais (por exemplo: Mortadelo é moi hábil para disfrazarse e para cabrear ó seu xefe). Tamén saberemos o seu enderezo e o número de teléfono de cada axente.

Ademais, a axencia está dividida en seccións, das que sabemos o seu código (único), o nome, unha breve descrición do seu cometido, e que axente é o director desa sección. Por suposto, cada axente estará nunha única sección, e nunha sección pode haber moitos axentes. Para acceder a cada sección hai unha serie de entradas secretas numeradas (este número será único para cada entrada dentro de cada sección), das que sabemos o seu emprazamento e o horario no que poden ser usadas.

Por outro lado, a axencia dispón dun catálogo co equipamento para realizar as súas misións. No catálogo hai tipos de equipamento que se identifican por un código único, e terán un nome e unha descrición. Podemos ter máis dunha unidade de equipamento de cada tipo de equipamento do catálogo, e cada unha destas unidades identificarase polo código do tipo de equipamento máis un número correlativo. Queremos saber, por suposto, a cantidade de unidades que temos de cada tipo de equipamento. Por exemplo, do equipamento de código XR-3 (Avión Fantasma) temos dúas unidades, identificadas por XR-3 número 1 e XR-3 número 2, mentras que das pistolas de auga A-1 temos 10 unidades, que van da A-1 número 1 á A-1 número 10.

Algunha destas unidades de equipamento, como o XR-3 (O avión fantasma) número 2, están asociados a unha única sección (neste caso, a de contraespionaxe), e queremos saber esa información. En cambio, hai outras unidades de equipamento, como calquera das pistolas de auga A-1, que non están asociados a ningunha sección.

Por suposto, o principal cometido desta axencia é levar a cabo misións secretas. Cada misión terá un código e un obxectivo a cumprir, unha data de inicio, e unha data de finalización, e o resultado final. Cada misión será levada a cabo por 1 ou máis axentes, e queremos saber cando empezou a traballar cada axente en cada misión (as datas dos distintos axentes poden ser distintas entre si, e non teñen que coincidir coas datas de inicio e fin da misión).

Ademais, algunhas misións requiren o uso de algunha unidade de equipamento por parte de algún axente. Se ese é o caso, queremos saber que axente usou cada unidade nesa misión, e o estado no que quedou a unidade tras a súa finalización. Este estado pode ser Intacto, Destrozado ou Perdido.

Exercicio: Aluguer de coches

Debemos modelizar a base de datos dunha empresa de aluguer de coches. A empresa ten múltiples oficinas ao longo do mundo. Cada oficina identifícase cun código (único entre tódalas oficinas do mundo), ten un enderezo, localidade, Tlf e país.

A empresa dispón dunha flota de coches. Cada coche ten unha matrícula, marca e cor, e pertence a un tipo (A, B, C, D, E, ou F). Cada tipo de coche ten unha descrición e pode alugarse con diferentes tarifas. Cada tarifa ten un código que a identifica, unha descrición e un prezo para cada tipo de coche. Así a tarifa 1 –por exemplo, “fins de semana”– existe para todos os tipos de coches e ten un prezo distinto para cada tipo de coches.

A clientela debe estar dada de alta na base de datos. De cada persoa que é cliente gardamos os seguintes datos: Número de identificación (nº pasaporte, DNI, etc.), nome, enderezo, localidade, país, nº tarxeta de crédito e tlf.

Cando se aluga un coche, codifícase o aluguer cun código de contrato, e ademais almacénase a persoa (cliente) que asina o contrato, a matrícula do coche que leva, a data e hora de retirada do vehículo, os litros de combustible e Kms que tiña, e a tarifa que se vai aplicar. Tamén se almacena a oficina onde se retira o coche.

Xa no momento do aluguer debemos almacenar a oficina, e a data e hora previstas en que se espera facer a devolución do coche. No momento da devolución debemos anotar a oficina, data e hora en que realmente se realizan (aínda que coincidan cos valores esperados que se indicaron ao alugar o vehículo). É dicir, débese gardar a data, hora e oficina de devolución previstas e as reais, para poder comparar as previsións co que realmente aconteceu. Tamén se almacena a kilometraxa e litros de combustible do coche no momento real da devolución.

Exercicio: Cadea de ximnasios

Queremos deseñar unha base de datos para unha empresa que xestiona unha cadea de ximnasios. Cada ximnasio identifícase por un código, e ten un nome e un enderezo.

En cada ximnasio hai unha ou máis dependencias ou salas, que se identifican por un número dentro de cada ximnasio, e das que sabemos ademais o nome que recibe a sala, e a súa capacidade.

Así, no ximnasio de código X01 temos a sala 1, que é a sala de musculación, con capacidade para 20 persoas, e a sala 2, que é a sauna, con capacidade para 10. No ximnasio X02 a sala 1 é a sala de aeróbic, mentras que a sala 2 é a sala de musculación.

Dispoñemos tamén de máquinas para a realización de exercicios. Cada máquina ten un código de barras para identificala, e temos anotado tamén a data de compra. As máquinas serán dun determinado tipo, que nos servirá para almacenar a información común das máquinas que son iguais. Cada tipo está codificado na base de datos, e saberemos tamén o seu nome e unha descrición. Só temos información dos tipos para os que temos algunha máquina. Cada máquina está ubicada nunha sala concreta dun ximnasio.

Por exemplo, as máquinas 8450001 e 8450002 foron compradas ambas o 15 de xaneiro de 2013. Ambas son do tipo EL001, de nome "Orbitrack 4 En 1", e que se describe como "Bicicleta elíptica, con 8 niveis de tensión".

Tamén xestionamos os socios da cadea de ximnasios. Cada socio identifícase por un número, e gardamos o seu nome, enderezo e teléfono. Queremos saber cada unha das veces que cada socio acude a un dos nosos ximnasios, gardando o instante no que entra (a data e hora) e tamén o instante no que sae.

Exercicio: ITV

O servizo de Inspección Técnica de Vehículos (ITV) pídenos que modelemos unha pequena base de datos a partir das seguintes especificacións:

- ▶ Cada vehículo identifícase polo número de bastidor. Ten, ademais, unha matrícula, unha marca e un modelo.
- ▶ Un vehículo será propiedade dunha única persoa. De cada persoa, saberemos o seu nome completo, enderezo, e un e-mail de contacto para a avisar de que ten que pasar a inspección dun dos seus vehículos (pode ter varios). Identificamos a cada persoa cun código único.
- ▶ O servizo de ITV ten unha serie de centros que poden realizar as inspeccións. Aparte dun código único, de cada centro sabemos o seu enderezo, e o nome do seu responsable.
- ▶ Cando se fai a inspección dun vehículo, identificaremos esta inspección cun número único. Saberemos, evidentemente, o vehículo que se leva a inspeccionar, e en que centro, así como a data e a hora na que se levou a cabo. Tamén se garda o resultado final da inspección, que pode ser “superada” ou “Non superada”.
- ▶ En calquera caso, unha inspección pode identificar problemas no vehículo. En cada inspección gardarase a lista numerada de fallos (será un número correlativo que empeza en 1 para cada inspección), e indicarase a severidade do fallo (leve, grave ou moi grave), a categoría (iluminación, elementos de seguridade do vehículo, contaminación, etc.) e unha descrición máis detallada do fallo.

Se unha inspección ten algún fallo grave ou moi grave, o resultado final será “Non superada”. Noutro caso, será “Superada”.

Exemplo: O vehículo A realiza a inspección 10134, e ten os seguintes fallos: 1 (grave)(contaminación)(A emisión de CO supera o límite legal), e 2 (leve)(iluminación)(luz de freno dereita fundida); O vehículo B realiza a inspección 10137, con fallo 1 (leve)(iluminación)(luz antinéboa fundida); O vehículo C realiza a inspección 10138 e non se detectan fallos. Con estes datos podemos saber que o vehículo A non superou a inspección, pero os vehículos B e C si o fixeron.

Exercicio: Autobuses

Queremos deseñar unha base de datos para unha compañía de autobuses de liñas regulares

- ▶ A compañía dispón dunha flota de autobuses. De cada un sabe a súa matrícula, a marca, o modelo, e o ano de compra.
- ▶ Existe unha plantilla de conductores para estes autobuses. De cada conductor temos o NSS, nome, enderezo, e os carnés que ten. Os conductores non teñen un autobús asignado, dependerá da viaxe que realicen.
- ▶ Como é unha compañía de liñas regulares, temos definidas unhas rutas. De cada ruta sabemos a orixe, o destino (só os nomes dos pobos ou cidades), que días da semana se fai, e a hora de saída e chegada previstas. Cada ruta identifícase cun código único asignado pola compañía.

Como unha simplificación, neste modelo consideramos que as rutas van dunha orixe a un destino, sen paradas intermedias. Tamén consideramos que unha ruta se fai como máximo 1 vez ó día.

- ▶ Cando se realiza unha viaxe dunha destas rutas, irá un conductor concreto nun autobús concreto. Rexistraremos cada viaxe, indicando a hora real de saída, a hora real de chegada, o número total de pasaxeiros transportados, e anotaremos se houbo ou non algún incidente na viaxe.

Variacións:

- ▶ Considera que, aparte do nome do pobo que pode ser orixe ou destino dunha ruta, sabemos máis información: as coordenadas (lonxitude e latitude), a provincia, e o número de habitantes.
- ▶ Considera que as rutas realizan paradas intermedias en distintos pobos ó longo da ruta.

Exercicio: Estacións de servizo

Queremos deseñar unha base de datos para unha empresa que xestiona unha cadea de estacións de servizo (gasolineiras). Cada gasolineira identifícase por un código, e ten un nome e un enderezo. En cada gasolineira hai un ou máis surtidores, que se identifican por un número dentro de cada gasolineira. De cada surtidor sabemos o ano de instalación.

Así, na gasolineira G01, o surtidor 1 foi instalado no 2000 e o surtidor 2 no 2007. Para a gasolineira G02, o surtidor 1 foi instalado no 2004.

As gasolineiras dispoñen de depósitos para o combustible. Cada depósito alimenta varios surtidores, e un mesmo surtidor aliméntase de varios depósitos. Cada depósito identifícase por un código único, e ten unha capacidade en litros para determinado tipo de combustible. Os combustibles están codificados, e de cada un sabemos o seu código identificador, o nome e o prezo actual por litro.

Por exemplo, os depósitos 8450001 e 8450002 teñen unha capacidade de 2300 litros. Ambos almacenan gasolina “súper”, que temos codificada como GSP002, e ten un prezo actual de 1,53 euros.

Finalmente, queremos saber que coches repostan nas nosas gasolineiras. De cada coche gardamos a matrícula, marca e modelo. Queremos saber cada unha das veces que un coche repostou nunha das nosas gasolineiras, gardando a data. Evidentemente, un coche pode repostar varias veces na mesma gasolineira pero, para simplificar, podes supoñer que ningún coche repostou 2 veces no mesmo día.

Exercicio: ONGs

Queremos deseñar unha base de datos para unha ONG que apadriña nenos.

Dos nenos que apadriñamos sabemos o seu nome e o país no que reside, e asignámoslle un código único para identificalo. Almacenamos tamén datos dos captadores da ONG, que son as persoas que buscan facer novos socios para a ONG. Cada captador ten un código único que o identifica, un nome, un enderezo, e un ou varios teléfonos de contacto.

Cando un captador consegue un novo socio (tamén chamado “padriño”), gardará o seu DNI, nome, enderezo postal, provincia, email, e un teléfono de contacto. Asígnaselle tamén un código identificador único. Queremos tamén que o modelo mostre cantos socios ten captado cada captador.

Un socio pode apadriñar un ou varios nenos, pero un neno só pode ter un único padriño (por simplicidade, supoñemos que un neno non vai ter máis dun padriño ó longo de toda a súa vida). O apadriñamento formalízase cunha especie de contrato, que ten unha referencia única, onde se especifica quen é o padriño, quen o apadriñado, a data de inicio, a contía mensual que vai aportar o padriño, e o código da conta onde se domiciliarán os recibos. As aportacións serán mensuais, e gardaremos a data e o importe de cada un destes recibos.

Por outro lado, estes captadores reciben unha pequena comisión cada vez que conseguen formalizar un contrato novo. Faremos pagos mensuais a estes captadores, de xeito que se indicará a data, o importe pagado, e os contratos ou apadriñamentos que motivaron ese importe. Non é relevante para este modelo como se realiza o cálculo da comisión sobre os contratos.

Exercicio: Accidentes de tráfico

Na base de datos da Dirección Xeral de Tráfico (DXT) queremos gardar a seguinte información sobre accidentes de tráfico:

De cada accidente gardamos: código (único para cada accidente), día, hora, estrada, punto quilométrico e un atestado (un texto). Nun accidente hai polo menos un vehículo implicado, pero pode haber varios. O resultado do accidente pode dar lugar a cero ou moitas vítimas.

De cada accidente gardamos cero ou varias causas posibles do accidente. Cada causa posible dun accidente ten un número, que empeza en 1 para cada accidente: así a primeira causa posible dun accidente concreto sempre ten o número 1, se hai unha segunda causa posible terá o número 2, e así sucesivamente. Ademais cada causa posible ten unha descrición e un vehículo causante do accidente.

Das vítimas gardamos un código, que é único para cada vítima, nome, apelidos, enderezo e teléfono. Obsévese que unha vítima pode selo de un ou varios accidentes. De cada vítima temos que saber as lesións que sufriu (campo de texto) en cada accidente.

De cada vehículo implicado gardamos a súa matrícula, marca, modelo, cor, e sentido de circulación (por ex: sentido Betanzos). Obsévese que un vehículo pode estar implicado en moitos accidentes, e en cada accidente o sentido de circulación probablemente sexa un valor distinto.

Nun accidente ademais pode (non sempre) intervir un equipo de atestados da DXT. Cada equipo de atestados ten un código (único), unha base (unha localidade) e un área asignada (por ex: comarca de A Coruña).

Cada equipo de atestados está composto por 2 gardas civís. De cada garda civil gardamos o seu NSS, DNI, nome, enderezo, teléfono, data de ingreso no corpo e a súa categoría.